STAFF PAPER
RAC-SP-183

NOVEMBER 1962

Three General Network Flow Problems and Their Solutions

by
Robert G. Busacker
Stephen A. Coffin
Paul J. Gowen

PREPARED FOR THE
DEPARTMENT OF THE ARMY
Contract No. DA 44-188-ARO-1

OPERATIONAL LOGISTICS DIVISION
STAFF PAPER RAC-SP-183
Published November 1962

Three General Network Flow Problems and Their Solutions

by
Robert G. Busacker
Stephen A. Coffin
Paul J. Gowen

RESEARCH ANALYSIS CORPORATION
6935 Arlington Road, Bethesda, Md., Washington 14, D.C.

Received for Publication
15 June 1962
Published
November 1962
by
RESEARCH ANALYSIS CORPORATION
6935 Arlington Road
Bethesda, Md.

FOREWORD

In this paper computational procedures that solve three general mathematical problems concerning optimal flows through networks are described. The presentation is directed toward readers interested in complete mathematical validation of these algorithms or in the mathematical details of their implementation.

The requirement for incorporation of such procedures arose in the context of RAC Study 24.1, and the work reported here represents one aspect of that study. The objective of the study is to design a computer-assisted planning procedure for determining US Army midrange requirements for technical-service troops and class IV materiel for specific campaigns. One of the tasks involved in the planning process is the detailed routing of supplies through highway and railroad networks. The material presented here provides a mathematical basis for efficiently selecting desirable routings through such networks.

The first of the three procedures determines minimum-cost flows between specified network nodes when a maximum flow rate or " capacity " is given for each link, and when the cost of flow in a link is proportional to the magnitude of flow. The second determines minimum-cost flows when minimum as well as maximum flow rates are stipulated for all links. The third determines whether or not a given flow pattern minimizes cost. In the event it does not, the procedure then transforms the given flow pattern into one that is optimal.

A computer program employing the first two of these procedures has been written for the IBM 7090 computer and constitutes one operational package in the new planning system. Readers interested in the applications of these results in the specific context of Study 24.1 are referred to ORO-T-393, "Computer-Assisted Strategic Logistic Planning: Transportation Phase"; ORO-SP-160, "Programming Computer-Assisted Transportation Planning"; and ORO-TP-45, "Computer-Assisted Preparation of Transportation Annex, Planner Manual." Further discussion, including documentation and use of the computer program, will appear in the final report of Study 24.1.

The material presented here constitutes an adaptation and unified presentation of various known results in network flow theory, in a form useful to support the above study.

CONTENTS

FOREWORD	iii
ABSTRACT	2
INTRODUCTION	3
GENERAL DEFINITIONS AND RESULTS	7
STATEMENT OF FLOW PROBLEMS A, B, AND C	23
A TRANSPORTATION APPLICATION EMBODYING PROBLEMS A, B, AND C	27
SOLUTION OF PROBLEM A	35
Procedure for Solving Problem A	42
SOLUTION OF PROBLEM B	57
Procedure for Solving Problem B	68
SOLUTION OF PROBLEM C	70
Procedure for Solving Problem C in Type A Networks	75
REFERENCES	77
GLOSSARY OF PRINCIPAL SYMBOLS	78
INDEX	79
FIGURES	
1. Illustration of Paths and Cycles	10
2. Examples of Flow Patterns	13
3. Decomposition of a Flow Pattern into Elements	19
4. Schematic of a Transportation Network	28
5. Relation between Networks (X, U) and $(\bar{X}, \bar{U})$	59

THREE GENERAL NETWORK FLOW
PROBLEMS AND THEIR SOLUTIONS

ABSTRACT

Computational procedures for solving three general network flow problems are presented, together with proofs establishing their validity. Two of the problems are concerned with the determination of feasible flows (i.e., flows that lie between prescribed bounds in every arc of the network) whose costs are minimum, where arc costs are proportional to the magnitude of arc flow. The third problem involves the determination of whether or not a given feasible flow has minimum cost. The utility of the three procedures is illustrated in the context of a general transportation application. The material presented constitutes an adaptation and unified discussion of various known concepts and results in network flow theory.

INTRODUCTION

This paper extends the methods described in ORO-TP-15[1] to a wider class of network flow problems. In that publication, networks that had a specified "capacity" as well as a "unit cost" associated with each direction of every edge or "link" were considered. For convenience, parallel links were excluded in order that a link might be uniquely identified by its two vertices. The problem under consideration concerned the determination of the maximum steady-state rate of flow that could be sustained between two specified vertices, and the determination of a specific family of minimum-cost flow patterns between these vertices—one pattern for every integral rate of flow from zero to the maximum. That problem is essentially Problem A of this paper. The restriction on parallel links has been removed here, and the problem and its solution are presented in terms more closely allied to the related Problems B and C.

The method adopted in ORO-TP-15[1] was one of discovering a succession of path flows from the origin to the destination in such a way that each new path flow contributed one more unit of flow with the least possible increase in cost. The mechanism for selecting the "best" path at each stage of the process consisted of associating an economic length (called the "effective length") with each direction of every link. This length was a function of the appropriate constant unit cost and also of the amount of flow already present in the link owing to paths previously selected. By defining these lengths properly the best path to use next was simply the shortest in terms of total economic length.

A vital point in the procedure was the fact that a starting feasible flow pattern having minimum cost was available. (This was simply the pattern having zero flow in every link and consequently zero total cost.)

An important extension of this problem arises if one wishes to "force" flow through one or more links. That is, the flow in certain links is required to be in a specified direction and to be of a magnitude lying between two positive bounds. In this case there may not be any feasible flow patterns if the constraints are too stringent. If any feasible flows between the specified origin and destination exist then there are integers m and M that represent the minimum and maximum feasible rates of flow between these points. One can again pose the problem of determining a family of minimum-cost flows, one for each integral rate of flow between m and M. This is Problem B of the present paper. One can deal readily with problems of this type by employing an appropriate modification of the network, whose effect is to remove the constraints that "force" flow, and hence to reduce the problem to one of the previous type. The procedure adopted here involves a network augmentation that differs slightly from that of Berge,[2] together with the necessary extension of ideas required to find a complete set of minimum-cost flows rather than simply one.

The basic combining properties of elementary flows (corresponding to paths and cycles), which play an important role in the solution of Problem A (and, indirectly, Problem B), also yield a criterion for testing the optimality, in terms of cost, of a given feasible flow pattern. Problem C of this paper concerns such a test. A systematic method for testing, and for reducing the cost of nonoptimal patterns, is incorporated in the resulting procedure.

Network flow problems have been the object of considerable attention in the literature. The list of references given at the end of this paper is minimal. Concepts and results from these particular references were used to form the basis for the present exposition. Berge,[2] Fulkerson,[3] Ford and Fulkerson,[4] and Iri[5] give procedures for finding minimum-cost feasible flow patterns. Of these Ford and Fulkerson and Iri find complete families of minimum-cost patterns transmitting increasing amounts of flow from origin to destination. The principle of developing "expanding" flow patterns by successive addition of flows along appropriate "unsaturated" paths is discussed by Dantzig and Fulkerson.[6] Pollack and Wiebenson[7] review a variety of algorithms for finding shortest routes through networks. The particular shortest-route algorithm employed here (as the repeated "inner loop" in the procedure for solving Problem A) is a "labeling" process patterned after those presented by Ford[8] and Ford and Fulkerson.[9] The cited publication by Jewell[10] includes a development of ideas similar to that adopted here, particularly for the solution of Problem A. Since a comprehensive canvass of the literature of flow theory was not undertaken, other related material besides that referenced here has appeared in the literature.

The next section of this paper presents both fundamental terminology and results required for the precise formulation and solution of the three flow problems considered. The following section states the three problems. A basic vocabulary having been established, the next section indicates how all three mathematical problems arise in the contest of a general transportation problem. (The reader who does not wish to have the continuity of the mathematical development broken may omit this section without loss of any essential mathematical facts.) The remaining three sections develop and justify the procedures for solving the three problems. A glossary of recurring symbols and an index of principal terms are included at the end.

GENERAL DEFINITIONS AND RESULTS

Let X and U denote two sets, whose elements will be referred to as vertices and arcs respectively. Assume that two distinct vertices $x \in X$ and $y \in X$ are associated with each arc $u \in U$, and are designated as the initial and terminal vertex of u, respectively. This relationship is expressed symbolically by writing $u \cong (x, y)$. This is read "arc u extends from vertex x to vertex y," because of the geometric interpretation described below.

The pair (X, U) together with the association of vertices with arcs is a directed graph. If X and U are both finite sets the graph is called a finite directed graph. A finite directed graph can always be represented geometrically in three-dimensional Euclidean space by a set of points corresponding to the vertices and a set of simple open curves* representing the arcs. Each curve is constructed so that its end-points are the points in space corresponding to the two vertices associated with the arc it represents. Moreover the curves can be constructed in such a

*A simple open curve is a point set topologically equivalent to (or homeomorphic with) a straight-line segment; i.e., it can be transformed into a segment by a suitable one-to-one transformation continuous in both directions.

manner that they are mutually disjoint, except for common end-points they may share. Finally, each curve is considered to be directed, the direction being induced by traversing the curve from its initial to its terminal vertex. (Actually, a much larger class of directed graphs than merely the class of finite graphs has geometric realizations of the above type. Any graph for which neither X nor U has a higher order of infinity than the set of real numbers can be represented.) Most of the terms and concepts associated with graphs are particularly transparent when related to geometric graphs, as structures of the above type are called.

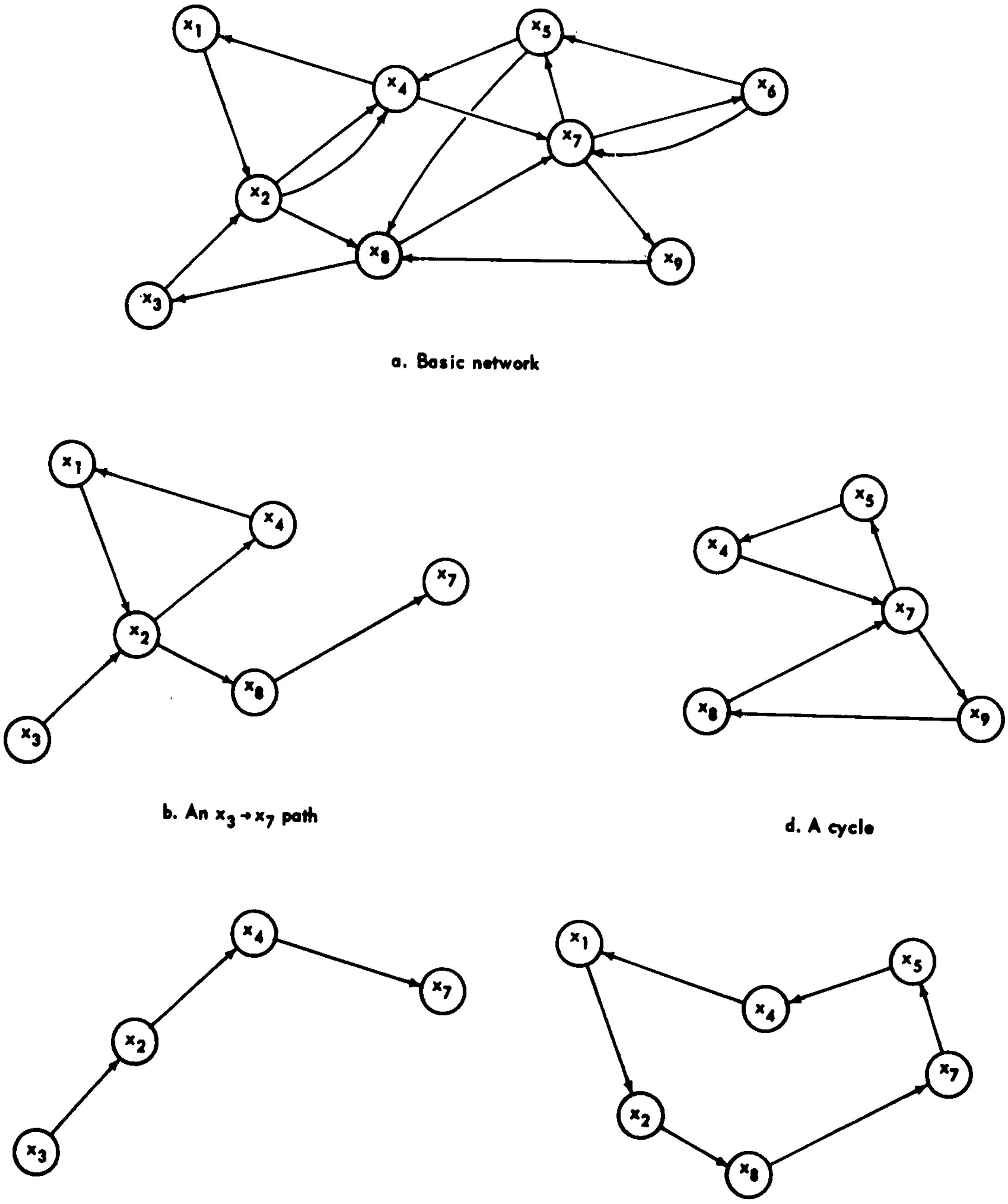
The initial and terminal vertices of an arc are collectively called its end-points. An arc is said to be incident with its end-points, and non-incident with all other vertices. A vertex that is not incident with any arc is an isolated vertex. If u_1 and u_2 are distinct arcs such that $u_1 \cong (x, y)$ and either $u_2 \cong (x, y)$ or $u_2 \cong (y, x)$ then u_1 and u_2 are termed parallel arcs. If $u_2 \cong (x, y)$ then u_1 and u_2 are strictly parallel arcs. Two distinct arcs are adjacent arcs if they have a common end-point. They are strictly adjacent arcs if the terminal vertex of one coincides with the initial vertex of the other. If u_1 and u_2 are strictly adjacent, and the terminal vertex of u_1 coincides with the initial vertex of u_2, u_1 is said to precede u_2.

A path is a set of arcs that can be arranged in a sequence $\{u_1, u_2, \ldots, u_n\}$ such that u_i precedes u_{i+1} for $i = 1, 2, \ldots, n - 1$ and such that the initial vertex of u_1 is distinct from the terminal vertex of u_n. More specifically, the above path is called an $x \rightarrow y$ path if x is the initial vertex of u_1 and y is the terminal vertex of u_n. In a geometric directed graph an $x \to y$ path corresponds to an open curve with end-points x and y such that if one envisions "traversing" it from x to y, traversing the arcs in the order, $u_1, u_2, \ldots, u_n$, then the direction of each arc agrees with the orientation induced by this traversal. A path is a <u>simple path</u> if no vertex is incident with more than two arcs of the path. A <u>cycle</u> is defined in the same manner as a path except that the initial vertex of u_1 coincides with the terminal vertex of u_n. A <u>simple cycle</u> is one such that no vertex is incident with more than two arcs of the cycle. Figure 1 illustrates these important substructures of a graph.

It will be convenient to consider also sets of arcs that differ from paths and cycles only in that certain of the arcs have the "wrong" orientation. If (X, U) is a directed graph let U' denote a set of elements that are in one-to-one correspondence with the arcs of U, and let u' denote the element of U' corresponding to $u \in U$. The elements of U' will also be considered as arcs whose end-points are specified by the relation $u' \cong (y, x)$ whenever $u \cong (x, y)$. The symbol W will always denote the enlarged set $U \cup U'$.*

Consider now the directed graph (X, W) obtained by enlarging (X, U) in the above manner. If a simple $x \to y$ path $\{u_1 u_2, \ldots, u_k\}$ is given in (X, W), in general some of the arcs will be members of U, and others of U'. Arcs of the former type are called <u>normal arcs</u>, relative to the

*$A \cup B$ denotes the union of sets A and B; i.e., the set of elements appearing in one or both sets.



a. Basic network

b. An $x_3 \rightarrow x_7$ path

d. A cycle

c. A simple $x_3 \rightarrow x_7$ path

e. A simple cycle

Fig. 1—Illustration of Paths and Cycles

path, and arcs of the latter type are called inverted arcs. Similarly, given a simple cycle in (X, W) the arcs can be classified as normal or inverted in the same way.

A directed graph (X, U) is said to be connected if for every pair of distinct vertices x and y there is at least one simple $x \rightarrow y$ path in the extended graph (X, W).

In the remainder of this paper consideration will be restricted to directed graphs of a special type. Specifically, a network is defined as a finite directed graph that is connected.

Let (X, U) be an arbitrary network. A flow pattern, or simply a flow in (X, U), is a function f that maps the set of arcs U into the set of integers. If $u \in U$, the integer f(u) is called the flow in arc u. If $u \cong (x, y)$ the flow is said to be "from x to y," or "toward y" and "away from x," if $f(u) > 0$. It is said to be "from y to x," or "toward x" and "away from y," if $f(u) < 0$.

This terminology is suggested by interpreting (X, U) as a set of essentially one-dimensional channels (arcs) interconnecting certain locations (vertices) where some homogeneous substance is flowing through each arc u at a steady rate f(u) , and where the sign of f(u), in conjunction with the orientation of u, indicates the direction of flow.

Let $U(\rightarrow x)$ denote the set of arcs of U having x as terminal vertex, and let $U(x \rightarrow)$ denote the set of arcs of U having x as initial vertex.

Given a flow pattern f the summation* $\Sigma_{U(x\rightarrow)}f(u)$ denotes the total magnitude of flow away from x, minus the total magnitude of flow toward x, contributed by arcs in the set $U(x\rightarrow)$. Similarly $\Sigma_{U(\rightarrow x)}f(u)$ denotes the total flow toward x, minus total flow away from x, for arcs of $U(\rightarrow x)$. The <u>net output at vertex x relative to flow pattern f,</u> denoted by $\Omega_f(x)$, is defined as follows:

$$\Omega_f(x) = \Sigma_{U(x\rightarrow)}f(u) - \Sigma_{U(\rightarrow x)}f(u) .$$

If either of the sets $U(x\rightarrow)$ or $U(\rightarrow x)$ is void the corresponding summation is taken to have value zero. (It is a consequence of the connectivity of (X, U) that not both sets can be void.)

One can classify the vertices of (X, U) in terms of the values of their respective net outputs as follows:

If $\Omega_f(x) > 0$, x is a <u>source</u>.

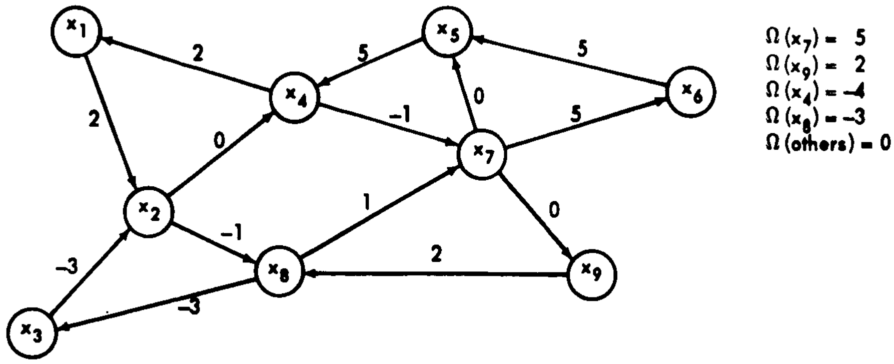
If $\Omega_f(x) < 0$, x is a <u>sink</u>.

If $\Omega_f(x) = 0$, x is an <u>intermediate vertex</u>.

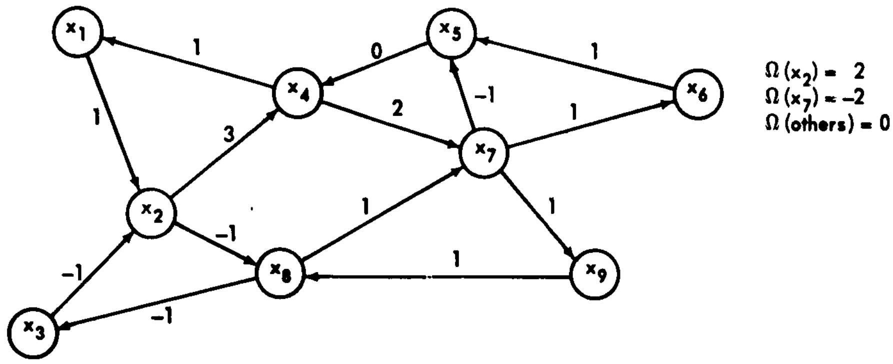
It is readily seen that $\Sigma_X \Omega_f(x) = 0$ since, when $\Sigma_X \Omega_f(x)$ is expanded, the resulting double summation involves $f(u)$ precisely twice for every arc $u \epsilon U$—once with a positive sign and once with a negative sign. [Since $u \epsilon U(x\rightarrow)$ for precisely one $x \epsilon X$, and $u \epsilon U(\rightarrow y)$ for precisely one $y \epsilon X$, and $y \neq x$.] Figure 2a illustrates the above notions.

If $\Omega_f(x) = 0$, except possibly for $x = x_i$ and $x = x_j$, then $\Omega_f(x_i) = - \Omega_f(x_j)$. So either every vertex is an intermediate vertex or else x_i and x_j are the

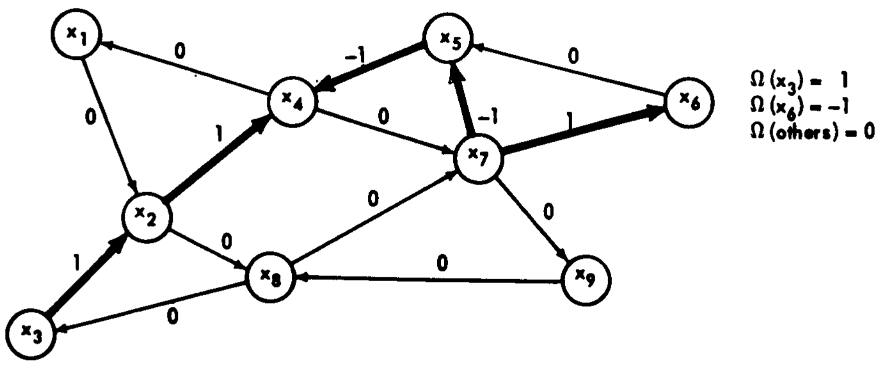
*Summations of the form $\Sigma_{u \epsilon U}$ will be written as simply Σ_U when ambiguity will not arise.



a. A flow pattern



b. An $x_2 \rightarrow x_7$ flow pattern of value 2



c. An elementary $x_3 \rightarrow x_6$ path flow

Fig. 2—Examples of Flow Patterns

source and sink, in some order. Assuming that x_i is the source, i.e., that $\Omega_f(x_i) > 0$, such a pattern is called an <u>$x_i \rightarrow x_j$ flow pattern.</u> Given an $x_i \rightarrow x_j$ flow pattern f, $\Omega_f(x_i)$ is called the <u>value</u> of the flow pattern. Intuitively, it represents the rate at which flow "moves" from x_i to x_j. It will be convenient to include, in the class of $x_i \rightarrow x_j$ flow patterns, patterns f also such that $\Omega_f(x) = 0$ for all $x \epsilon X$, including x_i and x_j. Such patterns will be considered as <u>zero-valued</u> $x_i \rightarrow x_j$ flow patterns.

If f and g are two flow patterns defined on the same network (X, U), f and g are said to be <u>conformal</u> if there is no arc $u \epsilon U$ such that $f(u) \cdot g(u) < 0$. Expressed differently, for every arc $u \epsilon U$, either $f(u) \geq 0$ and $g(u) \geq 0$ or else $f(u) \leq 0$ and $g(u) \leq 0$. A set $\{ f_1, f_2, \ldots, f_k \}$ of flows is conformal if the flows are pairwise conformal.

The <u>sum</u> of two flow patterns f and g, denoted by f + g, is defined by the relation $(f + g)(u) = f(u) + g(u)$ for all $u \epsilon U$. Similarly, the <u>difference</u> f - g is defined by the relation $(f - g)(u) = f(u) - g(u)$ for all $u \epsilon U$. If f_1 and f_2 are both $x_i \rightarrow x_j$ flow patterns having values k_1 and k_2 respectively, it is easily seen that $f_1 + f_2$ is also an $x_i \rightarrow x_j$ flow pattern whose value is $k_1 + k_2$. Moreover, $f_1 - f_2$ is a flow pattern of value $|k_1 - k_2|$, and is an $x_i \rightarrow x_j$ or $x_j \rightarrow x_i$ flow pattern if $k_1 - k_2 \geq 0$ or ≤ 0 respectively.

Two particularly simple types of flow patterns, called <u>elementary</u> flow patterns, play a central role in the following development. It will be seen that these constitute basic building blocks for more complex flow patterns in the sense that a given pattern can be decomposed into (expressed as the sum of) appropriate elementary patterns, and patterns of the types sought later can be synthesized from appropriate elementary patterns.

Let (X, U) be an arbitrary network, and let (X, W) denote the enlarged network described earlier where $W = U \cup U'$. If C is a simple $x \to y$ path in (X, W), an associated flow pattern f in (X, U) is defined as follows:

$$f(u) = \begin{cases} 1 \text{ if } u \in C \\ -1 \text{ if } u' \in C \\ 0 \text{ if neither } u \in C \text{ nor } u' \in C. \end{cases}$$

It is easily verified that f is an $x \to y$ flow pattern whose value is 1. Such flow patterns are called <u>elementary $x \to y$ path flows</u> (see Fig. 2).

If C is a simple cycle in (X, W), and f is defined in the same manner, it is easily seen that f is a zero-valued flow pattern, called an <u>elementary cycle flow</u>.

The following lemma establishes the fact that an arbitrary $x_1 \to x_2$ flow pattern can be expressed as the sum of conformal elementary flows.

<u>Lemma 1</u>

If f is an $x_1 \to x_2$ flow pattern of value k in a network (X, U), and $f(u) \neq 0$ for at least one arc $u \in U$ then it is possible to express f as a sum of the form

$$f = g_1 + \ldots + g_k + h_1 + \ldots + h_n$$

where the g_i's are k elementary $x_1 \to x_2$ path flows, the h_j's are elementary cycle flows, and the g_i's and h_j's are jointly conformal.

<u>Proof</u>. Assume first that $k > 0$. Since $\Omega_f(x_1) = k > 0$ there exists either an arc $u \in U(x_1 \to)$ such that $f(u) > 0$ or else an arc $v \in U(\to x_1)$ such that $f(v) < 0$. (Of course arcs of both types may exist.) Let u_1 denote an arc of either type, and let y_1 denote its other end-point. Since u_1

contributes flow toward y_1, and since $\Omega_f(y_1) = 0$ (assuming $y_1 \neq x_2$) there is either an arc $u \in U(y_1 \rightarrow)$ with $f(u) > 0$ or else an arc $v \in U(\rightarrow y_1)$ with $f(v) < 0$. Let u_2 be such an arc and let y_2 denote its other end-point. Continuing in this manner, one of two possibilities must eventually arise. A stage k may occur such that $\{ x_1, y_1, y_2, \ldots, y_{k-1}\}$ are all distinct vertices, but that $y_k = x_1$ or $y_k = y_j$ for some $j < k$. If this does not happen a stage k must occur eventually such that $y_k = x_2$, since there are a finite number of vertices. In either case let $\{ w_1, w_2, \ldots, w_k\}$ denote the sequence of arcs of W related to $\{ u_1, u_2, \ldots, u_k\}$ as follows: $w_i = u_i$ if $f(u_i) > 0$ (i.e., if y_i is the terminal vertex of u_i), while $w_i = u_i'$ if $f(u_i) < 0$ (i.e., if y_i is the initial vertex). If $y_k = y_j$ for some $j < k$ then $\{ w_{j+1}, w_{j+2}, \ldots, w_k\}$ is a simple cycle. Similarly, if $y_k = x_1$ the entire set of w_i's is a simple cycle. If the y_i's are distinct but $y_k = x_2$ then the entire set of w_i's forms a simple $x_1 \rightarrow x_2$ path.

Denote by l_1 the elementary cycle flow or elementary $x_1 \rightarrow x_2$ path flow corresponding to the cycle or path determined in this manner, and set $f_1 = f - l_1$. In the former case f_1 is again an $x_1 \rightarrow x_2$ flow pattern of value k. In the latter case f_1 is an $x_1 \rightarrow x_2$ flow pattern of value $k - 1$. In either case if the value of f_1 is greater than zero a second elementary flow l_2 can be "extracted" by repeating the process. Let $f_2 = f_1 - l_1 = f - l_1 - l_2$. In general, having determined $f_i = f - \sum_{j=1}^{i} l_j$ either f_i has value zero or else another elementary path or cycle flow l_{i+1} can be found so that one can define $f_{i+1} = f - \sum_{1}^{i+1} l_j$. But every time an elementary flow is subtracted the effect is to reduce the value of certain positive arc flows by one unit, increase the value of certain negative arc flows by one unit, and leave unchanged the flows in arcs not associated with the cycle or path. Such modifications can occur only a finite number of times before all arc flows are reduced to zero. So eventually a stage m is reached such that $f_m = f - \sum_1^m l_j$ is a zero-valued flow pattern z, and $\{l_1, l_2, \ldots, l_m\}$ consists of k elementary $x_1 \to x_2$ path flows and m - k cycle flows.

If z is identically zero the desired decomposition has been produced. If not, let y be any vertex such that $z(u) > 0$ for some $u \in U(y \to)$ or else $z(v) < 0$ for some $v \in U(\to y)$. Since $\Omega_z(x) = 0$ for all $x \in X$, one can again generate a sequence of arcs as before. In this case x_1 and x_2 do not play an exceptional role, and eventually the path determined by "tracing" flow through a succession of vertices must lead back to a vertex previously encountered. When this occurs, another cycle flow l_{m+1} is determined, yielding another zero-valued pattern $z_1 = z - l_{m+1}$. Either $z_1(u) = 0$ for all $u \in U$, or still another cycle flow l_{m+2} can be "extracted." Ultimately one obtains $Z = z - l_{m+1} - l_{m+2} \ldots - l_p$, where Z is identically zero. But then $f = \sum_1^m l_j + z = \sum_1^p l_j$ where k of the l_j's are path flows and p - k are cycle flows. Setting n = p - k the desired decomposition has been obtained. (If f was initially of value zero, only the part of the process related to decomposing z is required.)

The fact that the elementary flows are conformal follows from the nature of their selection. Arc flows were consistently chosen that agreed in direction with those of the original pattern f. If $f(u) \geq 0$ then $l_j(u) \geq 0$ for every j, while if $f(u) < 0$ then $l_j(u) \leq 0$ for every j. (More precisely in the former case $l_j(u) = 1$ for f(u) of the elementary flows and $l_j(u) = 0$

for the others. In the latter case $l_j(u) = -1$ for $-f(u)$ elementary flows and $l_j(u) = 0$ for the rest.) This completes the proof.

In order to clarify the foregoing procedure Fig. 3 illustrates two decompositions of a flow pattern of value 2 into conformal elementary flows. (It should be noted that such decompositions are not in general unique.)

A network (X, U) is said to be capacitated if two integer-valued functions b and c are defined on U and satisfy the following relation:

$$c(u) \geq 0 \text{ and } c(u) \geq b(u) \text{ for all } u \in U \text{ .}$$

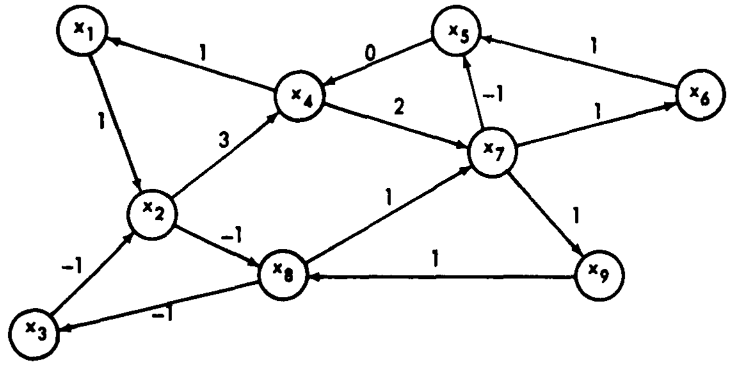
The integers $c(u)$ and $b(u)$ are called the upper and lower bounds on flow in arc u respectively. A flow pattern f is feasible in arc u if $b(u) \leq f(u) \leq c(u)$. If $b(u) < 0$ this means that the set of feasible flows in u range from $c(u)$ units of flow in the direction associated with u to $-b(u)$ units of flow in the reverse direction. If $b(u) > 0$ one says that the flow in u is "forced" to be in the direction of arc u, and to range from a minimum of $b(u)$ to a maximum of $c(u)$ units of flow. A flow pattern is feasible in a network if it is feasible in every arc of the network.

The following lemma concerning feasible flow patterns will be required in later sections.

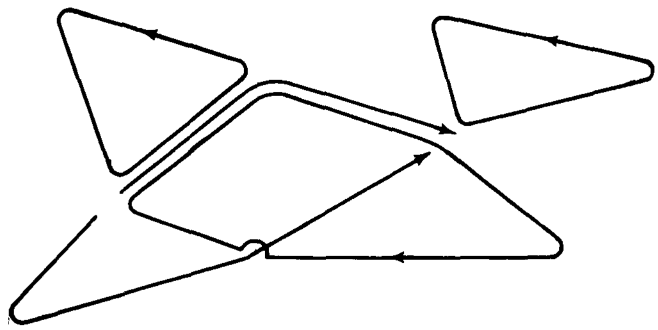
Lemma 2

Let f_1 and f_2 be feasible $x \to y$ flow patterns having values k_1 and $k_2 \leq k_1$; and let the $x \to y$ flow pattern $f_1 - f_2$ be written as a sum of conformal elementary flows:

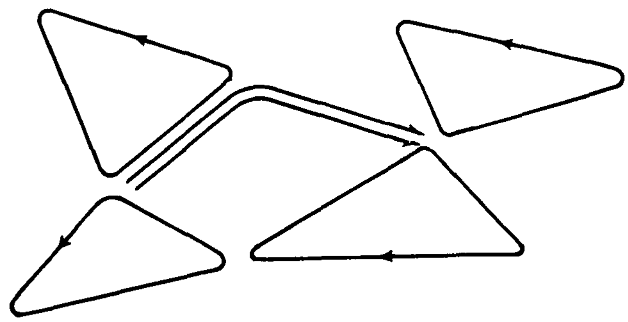
$$f_1 - f_2 = g_1 + \ldots + g_k + h_1 + \ldots + h_n$$



a. An $x_2 \to x_7$ flow pattern of value 2



b. Decomposition into elementary flows†



c. Alternate decomposition †

Fig. 3—Decomposition of a Flow Pattern into Elements

†In b and c, arrows indicate direction of flow, not arc orientations as in a.

where $k = k_1 - k_2$. Then $f_2 + S$ is also a feasible $x \to y$ flow if S is the sum of any subset of $\{ g_1, \ldots, g_k, h_1, \ldots, h_n \}$.

Proof. Since the g_i's and h_j's are jointly conformal, in any arc u they are all nonnegative if $f_1(u) \geq f_2(u)$ or all nonpositive if $f_1(u) < f_2(u)$. In either case adding a subset of them to f_2 yields an arc flow f(u) that lies between the values $f_1(u)$ and $f_2(u)$. Since $f_1(u)$ and $f_2(u)$ each lie between b(u) and c(u) it follows that f(u) does also. Hence f is a feasible flow pattern. This completes the proof.

A network is said to be weighted if two functions a_1 and a_2 called unit cost functions are defined on the set of arcs, whose values are nonnegative real numbers. If u is any arc, $a_1(u)$ is considered the unit cost of flow in the direction of the arc, and $a_2(u)$ is the unit cost of flow in the opposite direction. If f is a flow pattern the cost of flow in u, denoted by $T[f(u)]$, is defined as $a_1(u) \cdot f(u)$ if $f(u) \geq 0$, and as $-a_2(u) \cdot f(u)$ if $f(u) < 0$. Thus in any case the cost of flow in an arc is nonnegative. The total cost of a flow pattern f is denoted by T(f). (Two cost functions are specified in order that asymmetric unit costs may be reflected.) The problems investigated in the remainder of this paper deal with constructing feasible flow patterns whose costs are minimum, in a given capacitated and weighted network. [Of course if $b(u) \geq 0$, $a_2(u)$ is irrelevant, since no feasible arc flow is such that $f(u) < 0$.] Feasible patterns whose costs are minimum will be called ideal patterns.

An important relation between the costs of certain flow patterns is given in the following result.

Lemma 3

If f, g, and h are $x \rightarrow y$ flow patterns in the same network, and g and h are conformal, then

$$T(f+g+h) - T(f+g) \geq T(f+h) - T(f) .$$

Expressed differently the incremental cost resulting from adding h to f + g is greater than or equal to the incremental cost resulting from adding h to f alone.

Proof. Let u denote an arbitrary arc of U, and set

$$T_u = T[f(u)+g(u)+h(u)] - T[f(u)+g(u)] - T[f(u)+h(u)] + T[f(u)] .$$

One can also write T_u as

$$T_u = |f(u)+g(u)+h(u)|\, a_i(u) - |f(u)+g(u)|\, a_j(u) - |f(u)+h(u)|\, a_k(u) + |f(u)|\, a_m(u),$$

where $a_i(u)$ is $a_1(u)$ or $a_2(u)$ depending on whether $[f(u)+g(u)+h(u)]$ is ≥ 0 or < 0, and a_j, a_k, and a_m are similarly determined. It will be shown that $T_u \geq 0$ for all $u \epsilon U$.

Suppose that $f(u) \geq 0$. [If $f(u) < 0$, u could simply be reoriented and $f(u)$ could be replaced by $-f(u)$ without essentially changing the flow pattern. Of course then $g(u)$ and $h(u)$ must also be replaced by their negatives and $a_1(u)$ interchanged with $a_2(u)$.] Since g and h are conformal, one of the following sets of inequalities must hold:

$$f(u) \geq 0,\ g(u) \geq 0, \text{ and } h(u) \geq 0 ,$$

or

$$f(u) \geq 0,\ g(u) \leq 0, \text{ and } h(u) \leq 0 .$$

If the first set of inequalities holds, then the absolute value signs are superfluous in the expression for T_u, and $a_i = a_j = a_k = a_m = a_1$. By evaluating T_u, $T_u = 0$ is obtained. If the second set of inequalities holds, but $[f(u) + g(u) + h(u)] \geq 0$, then $[f(u) + g(u)] \geq 0$ and $[f(u) + h(u)] \geq 0$, and $T_u = 0$ is again obtained. The remaining case is that for which $f(u) \geq 0$, $g(u) \leq 0$, $h(u) \leq 0$, and $f(u) + g(u) + h(u) < 0$. In this case, T_u can be written as

$$T_u = -[f(u)+g(u)+h(u)]\, a_2(u) - \left|f(u)+g(u)\right| a_j(u) - \left|f(u)+h(u)\right| a_k(u) + f(u)a_1(u).$$

Depending on the relative magnitudes of $f(u)$ and $g(u)$, $\left|f(u) + g(u)\right| a_j(u)$ is equal to either $[f(u) + g(u)]\, a_1(u)$ or $-[f(u) + g(u)]\, a_2(u)$. Similarly $\left|f(u) + h(u)\right| a_k(u)$ can be expressed either as $[f(u) + h(u)]a_1(u)$ or else as $-[f(u) + h(u)]a_2(u)$. By substituting these expressions in the above equation for T_u in all (four) possible combinations and simplifying the resulting expression, an expression for T_u that is clearly ≥ 0 is always obtained. Hence for every $u \in U$, $T_u \geq 0$. Hence $\Sigma_U T_u \geq 0$. But this is equivalent to the assertion of the lemma, and hence the proof is complete.

STATEMENT OF FLOW PROBLEMS A, B, AND C

In this section three specific network flow problems are stated. These are designated as Problems A, B, and C. The remainder of the paper is devoted to the presentation of complete solutions to these problems.

As common assumptions for all three problems it is assumed that a capacitated and weighted network (X, U) is given. Two vertices x_1 and x_n are distinguished from the remaining vertices, and are to be considered as the unique source and unique sink, respectively, of all flow patterns of interest. In other words the class of flow patterns under consideration is limited to the class of $x_1 \rightarrow x_n$ flow patterns. Retaining the symbols introduced earlier, b(u) and c(u) denote the lower and upper bounds of flow in arc u, and $a_1(u)$ and $a_2(u)$ the unit costs appropriate to the two directions that flow may assume in arc u.

For Problem A it is further assumed that $b(u) \leq 0$ for every $u \in U$. For such a network, called a Type A network, it is evident that a feasible $x_1 \rightarrow x_n$ flow pattern having value 0 exists, viz., the pattern defined by $f(u) = 0$ for all $u \in U$. This pattern is in fact ideal, since $T(f) = 0$ and $T(g) \geq 0$ for every feasible flow pattern g. It follows from Lemma 2 that if a feasible $x_1 \rightarrow x_n$ flow pattern of value k exists in a Type A network, then a feasible $x_1 \rightarrow x_n$ flow pattern exists having value i for $i = 1, 2, \ldots, k$. Since there are at most a finite number of feasible flow patterns because all arc bounds are finite and arc flows must be integers, this implies that there exist ideal $x_1 \rightarrow x_n$ flow patterns having values $1, 2, \ldots, k$. The largest k for which feasible $x_1 \rightarrow x_n$ flow patterns exist will be called the $x_1 \rightarrow x_n$ capacity of the network. In view of these considerations the following problem is always meaningful:

Problem A. Given a network (X, U) that is weighted and capacitated, and such that $b(u) \leq 0$ for all $u \in U$; develop a finite procedure that will (a) determine the $x_1 \rightarrow x_n$ capacity M of the network, and (b) determine a specific ideal $x_1 \rightarrow x_n$ flow pattern having value i for $i = 1, 2, \ldots, M$.

By a "finite procedure" is meant one that, for every network of the type under consideration, will yield a solution after a finite number of "simple" steps. A simple step is an elementary arithmetical operation or comparison of magnitudes performable by a digital computer. Although the procedure given here will not be broken into ultimate operations it will be given in sufficient detail so that this could readily be done.

Problem A is discussed and solved in full detail in ORO-TP-15.[1] A restatement of the solution procedure and some of its important characteristics is given in this paper because of the intimate relation of this problem with Problems B and C.

The assumptions for Problem B differ from those of Problem A in only one respect: it is assumed that $b(u) > 0$ for at least one arc $u \in U$. Networks of this type are called Type B networks. Thus there is at least one arc in which the flow is required to be in a specified direction, viz., the direction associated with the arc. It is easy to construct networks of this type for which no feasible flow, of any value, exists. Suppose, for example, that $U(\rightarrow x_1)$ and $U(x_n \rightarrow)$ are empty sets, so that no arc is directed toward the source or away from the sink. Then if $\Sigma_{u \in U(x_1 \rightarrow)} b(u) > \Sigma_{u \in U(\rightarrow x_n)} c(u)$ no feasible $x_1 \rightarrow x_n$ flow pattern is possible because the first summation is a lower bound on the value of flow, whereas the second summation is an upper bound. However, if any feasible $x_1 \rightarrow x_n$ flow patterns exist, then it will be shown that there exist minimum and maximum values, m and M, such that feasible flows exists for every value i such that $m \leq i \leq M$. The following problem is therefore always meaningful:

Problem B. Given a network (X, U) that is weighted and capacitated, and such that $b(u) > 0$ for at least one arc $u \in U$; develop a finite procedure that will (a) determine whether any feasible $x_1 \rightarrow x_n$ flow patterns exist, (b) determine m and M if any feasible patterns exist, and (c) determine a specific ideal $x_1 \rightarrow x_n$ flow pattern having value i for $i = m, m+1, \ldots, M$.

Problem C differs from the preceding ones in that it is concerned with ideal and nonideal $x_1 \rightarrow x_n$ flow patterns having the same value. Since no restrictions are placed on the sign of b(u), as in Problems A and B, the problem is posed for both Type A and Type B networks.

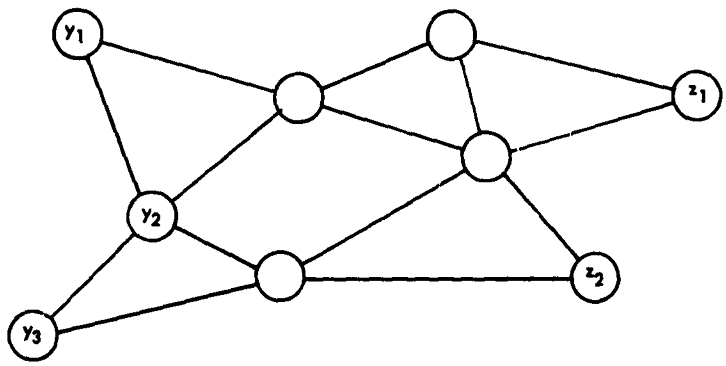
Problem C. Given a network (X, U) that is weighted and capacitated, and given a feasible $x_1 \rightarrow x_n$ flow pattern f_0 of value k; develop a finite procedure that will (a) determine whether or not f_0 is ideal, and (b) if f_0 is not ideal, find an appropriate succession of transformations, starting with f_0, each of which reduces the cost of the preceding flow pattern while retaining its value and feasibility.

A final note on the three problems is appropriate. A reasonable solution to each problem must consist of something substantially more efficient than an exhaustive procedure for generating and comparing all possible feasible flow patterns. Such a procedure would in fact be finite, but it is intended here that a procedure be devised that may be reasonably implemented.

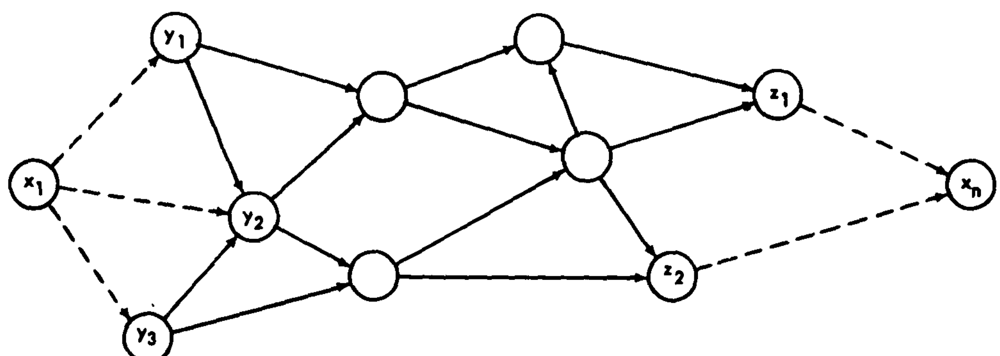
A TRANSPORTATION APPLICATION EMBODYING PROBLEMS A, B, AND C

This section considers a general class of problems arising in the field of transportation—problems having to do with the determination of "optimal" steady-state flows of materiel between certain origins and destinations interconnected by a transportation network. Problem A is presented as the mathematical formulation of the basic problem to be solved. Problem B then arises in connection with the attempt to gain control of the "continuity" of a succession of related problems of Type A. Finally, problems of Type C arise when one wishes to alter some of the parameters of the problem and determine whether or not optimality is disturbed, and (if it is) to restore optimality in a computationally efficient manner. The mechanics of solving Problems A, B, and C will not enter into the discussion. It is simply assumed here that efficient algorithms are available.

For the sake of concreteness suppose that one is concerned with the problem of moving goods from three locations y_1, y_2, and y_3 to two locations z_1 and z_2 through a highway network whose configuration is shown in Fig. 4a. (Although a very simple situation is depicted here, it



a. Basic network



b. Augmented network

Fig. 4—Schematic of a Transportation Network

should be noted that in practice the relevant network may have thousands of arcs. It is in such situations that the need for powerful general algorithms becomes apparent.)

Suppose that destinations z_1 and z_2 must receive R_1 and R_2 tons/day respectively, and that origins y_1, y_2, and y_3 can send at most S_1, S_2, and S_3 tons/day respectively. It is assumed that $S_1 + S_2 + S_3 \geq R_1 + R_2$, so that it is feasible to support the requirements at the destinations unless bottlenecks in the highway system prohibit sufficient flow.

As a constraint on the highway system itself, assume that each link joining two successive junction points has associated with it an integer representing the maximum rate of flow, in tons per day, that can be sustained during the time interval being analyzed. It is assumed that these link "capacities" apply in either direction and that they allow for retrograde movement of vehicles. (For the application considered here, only the "forward" movement of goods from origins to destinations is considered explicitly. Retrograde movement is assumed to retrace the forward routes.)

As a criterion for optimality, assume that it is desirable to determine a movement pattern which minimizes total ton-mileage per day. Thus the "unit cost" associated with each link is simply its physical length measured in miles, so that the "cost" of flow in a link has the dimensionality of ton-miles per day.

The problem is then one of determining the direction and rate of flow in each link so that (a) the net input to destination z_i is precisely R_i, (b) the net output from origin y_j is at most S_j, (c) the net output is zero at all intermediate junction points, (d) the flow in each link does not exceed its specified capacity, and (e) for all possible flow systems satisfying a to d the one selected involves minimum total ton-mileage of movement per day.

This problem can be reformulated as one involving a single origin (source) and destination (sink) by proper augmentation of the highway system to include certain hypothetical links. Figure 4b shows the configuration of the augmented system. A hypothetical source x_1 is added, together with arcs directed from x_1 to y_1, y_2, and y_3 respectively. These are assigned zero unit cost (or length) and capacities equal to the corresponding capabilities S_1, S_2, and S_3. Similarly a hypothetical sink x_n is added, with arcs directed from z_1 and z_2 to x_n. These also are assigned zero length and capacities equal to R_1 and R_2 respectively. Note that the links of the original highway system have also been oriented, so that this finite, connected system of vertices and arcs forms an $x_1 \rightarrow x_n$ network in the technical sense. The orientations here are for frame-of-reference purposes, so that the direction of flow may be indicated by the algebraic sign. (Although the direction of flow can usually be predicted beforehand for most arcs, in some of the lateral arcs it is not always clear which direction an optimal pattern will require.)

For arcs of the original network, set $b(u) = -K$ and $c(u) = K$, where K is the capacity of the corresponding link, and set $a_1(u) = a_2(u) = L$, where L is the length of the link. For arcs of the form $v \cong (x_1, y_i)$ set $b(v) = 0$, $c(v) = S_i$, $a_1(v) = 0$, and $a_2(v) = 0$. For arcs of the form $w \cong (z_j, x_n)$ set $b(w) = 0$, $c(w) = R_j$, $a_1(w) = 0$, and $a_2(w) = 0$.

Since b, c, a_1, and a_2 have been defined for the entire $x_1 \rightarrow x_n$ network, and the network is of Type A (since $b(u) \leq 0$ for all u), Problem A can be posed for this network. The $x_1 \rightarrow x_n$ capacity M is clearly such that $M \leq R_1 + R_2$, since flow can enter x_n only through the two arcs that join z_1 and z_2 to x_n, and these have capacities R_1 and R_2 respectively. Hopefully the solution of Problem A will reveal that $M = R_1 + R_2$, so that the requirements can in fact be feasibly fulfilled. Even if they cannot the maximum-valued ideal flow pattern will indicate the maximum tonnage that can be feasibly transmitted per day, and will further indicate a specific pattern of arc flows that realize this maximum tonnage with minimum cost (ton-mileage per day). The flows in a hypothetical arc should, of course, be interpreted as the amount of materiel sent from an origin or received by a destination, whichever is appropriate. Introduction of these arcs is merely an artifice that enables one to treat y_1, y_2, y_3, z_1, and z_2 as intermediate vertices for which input equals output, rather than as five exceptional vertices with distinctive constraints. Since the artificial arcs were assigned zero unit costs the cost of an ideal flow pattern in the augmented network is the same as the cost of the "real" flow, viz., the flow pattern obtained by finally disregarding the artificial arcs.

Suppose now that one is faced with the problem of determining optimum routings for several successive time intervals. For example, suppose that during month 1 the average daily demands at z_1 and z_2 are R_1 and R_2, and that during month 2 they change to R_1' and R_2' respectively. One can consider each month independently as a problem of Type A in the augmented network described above. However, it may well happen that the optimum flow patterns determined for the two months display a serious lack of "continuity." For instance, one pattern may rely heavily on northern routes and the other on southern routes. Or one may rely mostly on y_1 and y_2 as sources of supply and the other largely or perhaps exclusively on y_2 and y_3.

Assume that a specific arc is relied on heavily in the optimal solution obtained for month 1, and that the arc is not utilized in month 2. Although the solutions individually satisfy the criterion of minimum ton-mileage per day, taken together they may yield an unsound operational solution in the light of other considerations not reflected in the mathematical model. For instance, the use of this arc in month 1 may imply considerable investment in terms of road or bridge rehabilitation if the problem arises in a military context. To rehabilitate the link and then abandon it in a later month may well be unreasonable in view of the overall problem.

As a first measure month 1 might be reanalyzed, leaving all factors unchanged except for the elimination of the arc in question. Suppose that this is done, and that the new solution is not acceptable, either because the minimum ton-mileage per day is unacceptably higher than it was before or else because removal of the arc creates a bottleneck preventing satisfaction of the total demands at z_1 and z_2. In any event assume that an analysis of the situation leads to the conclusion that this highway link must be rehabilitated and used in month 1, and that it is highly desirable to continue its utilization in month 2, thus lessening reliance on other portions of the highway system.

One way to handle this problem is to reanalyze month 2, forcing flow through the relevant arc. In other words a Type B problem can be posed that differs from the original Type A analysis of month 2 only in that $b(u) > 0$ for this arc and a certain minimum flow is required. Assume that this analysis is carried out, obtaining a flow pattern f' with cost (ton-mileage per day) T(f'), which can be compared with the original month 2 solution f having cost T(f).

Since the month 2 problem was modified only in the direction of adding additional constraints, clearly $T(f') \geq T(f)$. It can happen that the new solution, forced to involve the arc in question, is as cheap, in terms of ton-mileage per day, as the old one. This can occur because the ideal flow pattern having a given value is not necessarily unique, so that although the original analysis of month 2 yielded an ideal pattern that did not involve the arc, another ideal pattern may actually exist that does. In any case a good "hard" number $T(f') - T(f)$ is obtained, which is a measure of the additional cost incurred if preserving continuity of flow in the arc is insisted on. This is just one example of how a Type B problem can be posed to "gain control" of a set of solutions to Type A problems when the initial solutions are not sufficiently compatible in terms of important constraints that were not initially built into the formulation.

It is sometimes also profitable to formulate problems of Type C in the context of a general transportation application of the above type. Suppose that there is serious doubt concerning the accuracy of some of the data. As an example, suppose that there is good reason to question the assumed capacity of an arc, and that this arc appears importantly in the solution as it now stands. Letting $c(u)$ be the capacity used in the analysis, suppose that the optimal pattern saturates this arc, so that $f(u) = c(u)$. Assume further that there is now reason to believe that $2c(u)$ is a better estimate of the arc's capacity, either because the original estimate was bad or else because appropriate physical measures can in fact double the capacity without undue expense. One approach would be to start from scratch, posing a new Problem A that differs from the original analysis only in that the capacity of u is increased. Another possibility, and one that may save a great deal of effort, arises by taking the original solution f, known to be optimal before, and inquiring whether or not it remains optimal when $c(u)$ is doubled. Clearly f is still a feasible pattern, which may or may not be optimal. Studying this Type C question, either f will emerge optimal in the modified problem, or else f can be used as the starting point for finding an ideal pattern f'. Here again a quantitative measure $T(f) - T(f')$ is obtained of the difference it makes whether $c(u)$ or $2c(u)$ is the capacity of u.

The preceding remarks are only intended to be suggestive of a host of problems that can be studied by using formulations of Types A, B, and C singly or in combinations. Network flow theory is applicable to many problems, including problems in which the network is itself entirely an artifice and has no physical counterpart. Where the network does exist, as in transportation problems, it is especially easy to formulate the appropriate mathematical questions because the model is so direct a reflection of reality. If the resulting questions are of any of these three types the methods of this paper or comparable methods are applicable.

SOLUTION OF PROBLEM A

In this section a résumé of the procedure for solving Problem A that appeared earlier in ORO-TP-15[1] is given, together with the intermediate mathematical results required to justify the procedure. This information is repeated here because of the intimate relation between Problems A, B, and C. It will be seen that the solution to Problem B involves solution of Problem A in a related network and that the solution of Problem C requires application of the central ideas that were needed for solving Problem A.

Recall that a Type A network is a weighted and capacitated network (X, U) such that $b(u) \leq 0$ for every arc $u \in U$.

It was noted earlier that a Type A network has the property that there exists an integer M, called its $x_1 \rightarrow x_n$ capacity, such that feasible $x_1 \rightarrow x_n$ flow patterns exist having values 0, 1, 2, . . ., M, but none exist having values greater than M.

Let (X, W) be the enlarged network obtained by introducing the set of "reverse" arcs U' in the manner described earlier. If f is a feasible $x_1 \rightarrow x_n$ flow pattern in (X, U) and P is a simple $x_1 \rightarrow x_n$ path or a simple cycle in (X, W), P is said to be <u>saturated</u> relative to f if it contains an arc $u \in U$ such that $f(u) = c(u)$ or an arc $u' \in U'$ such that $f(u) = b(u)$.

Otherwise, P is considered <u>unsaturated</u> relative to f. If g_P is the elementary $x_1 \to x_n$ path flow or cycle flow corresponding to P then it follows that $f + g_P$ is feasible if and only if P is unsaturated relative to f.

The following result provides the basis for a general procedure for finding feasible patterns having values 1, 2, . . ., M.

<u>Lemma 4</u>

If f_i is a feasible $x_1 \to x_n$ flow pattern in (X, U) having value i, and $i < M$ where M is the $x_1 \to x_n$ capacity of the network then there exists an unsaturated simple $x_1 \to x_n$ path in (X, W). If $i = M$ every simple $x_1 \to x_n$ path is saturated.

<u>Proof.</u> The last statement follows immediately. For assume $i = M$ and P is an unsaturated simple $x_1 \to x_n$ path. Then $f + g_P$ is feasible, where g_P is the corresponding elementary $x_1 \to x_n$ flow. But $f + g_P$ has value $i + 1 = M + 1$, contradicting the assumption that M is the $x_1 \to x_n$ capacity of (X, U).

Assume now that $0 \leq i < M$, and that f_i is a feasible $x_1 \to x_n$ flow pattern of value i. Let f_{i+1} be any feasible $x_1 \to x_n$ flow pattern of value i+1. Then $f_{i+1} - f_i$ is an $x_1 \to x_n$ flow pattern of value 1. According to Lemma 1

$$f_{i+1} - f_i = g_1 + h_1 + h_2 + \dots + h_n ,$$

where g_1 is an elementary $x_1 \to x_n$ path flow. But from Lemma 2 it follows that $f_i + g_1$ is feasible. Hence g_1 must correspond to an unsaturated path in (X, W). This completes the proof.

Thus if one has an algorithm for finding an unsaturated simple $x_1 \to x_n$ path relative to a given feasible $x_1 \to x_n$ flow pattern, whenever one exists, the following procedure will generate feasible patterns for all possible values:

(a) Let f_0 be the feasible $x_1 \to x_n$ flow pattern of value zero defined by $f_0(u) = 0$ for all $u \in U$.

(b) In general, having determined a feasible pattern f_i of value i, apply the algorithm to find (if possible) an unsaturated $x_1 \to x_n$ path P.

(c) If no such path P exists, i is the $x_1 \to x_n$ capacity of the network, and $\{ f_0, f_1, \ldots, f_i \}$ is a complete set of feasible $x_1 \to x_n$ patterns.

(c') If P is an unsaturated $x_1 \to x_n$ path, let g_P be the corresponding elementary path flow. Then $f_{i+1} = f_i + g_P$ is a feasible pattern of value $i + 1$, and step (b) can be repeated with f_{i+1} in place of f_i.

The solution of Problem A revolves around the fact that if care is taken in the above procedure to always select the "best" unsaturated path at each stage then the sequence $\{ f_0, f_1, \ldots, f_M \}$ will consist of ideal flow patterns. The following property possessed by ideal flow patterns can be established.

Lemma 5

If f_i is an ideal $x_1 \to x_n$ flow pattern of value i, and $i < M$, then an elementary $x_1 \to x_n$ path flow g exists such that $f_{i+1} = f_i + g$ is an ideal $x_1 \to x_n$ flow pattern of value $i + 1$.

Proof. Let f_{i+1} be any ideal $x_1 \to x_n$ flow pattern of value $i + 1$. Such a pattern must exist since $i < M$. Then $f_{i+1} - f_i$ is an $x_1 \to x_n$ flow pattern, which can (using Lemma 1) be written as

$$f_{i+1} - f_i = g_P + h_1 + \ldots + h_n ,$$

where g_P is an elementary path flow corresponding to some simple $x_1 \to x_n$ path P, and g_P is conformal with the elementary cycle flows $h_1, h_2, \ldots, h_n$. For convenience set $h = \Sigma h_i$, so that

$$f_{i+1} = f_i + g_P + h ,$$

where h is now a zero-valued flow pattern conformal with g_P. From Lemma 2 it is known that $f_i + g_P$ and $f_i + h$ are feasible flow patterns whose values are i + 1 and i respectively. Lemma 3 can be applied to obtain

$$T(f_i + g_P + h) - T(f_i + g_P) \geq T(f_i + h) - T(f_i) .$$

Since f_i is ideal the right-hand side is nonnegative. Hence

$$T(f_i + g_P + h) \geq T(f_i + g_P) .$$

But $f_i + g_P + h = f_{i+1}$. Since f_{i+1} is also ideal, equality must hold in the preceding expression. So $f_i + g_P$ also minimizes T, i.e., $f_i + g_P$ is ideal for value i + 1. This completes the proof.

Lemma 5 is of fundamental importance for the attack on Problems A, B, and C adopted here, and hence some amplifying remarks concerning its significance are in order. If f_i is ideal but not maximal, i.e., if $i < M$, all possible feasible patterns of value i + 1 can be obtained from f_i by adding appropriate patterns of value 1. Specifically, if f_{i+1} has value i + 1, one merely adds $(f_{i+1} - f_i)$ to f_i. Although there are a finite number of distinct feasible patterns of value i + 1, for nontrivial problems in complex networks the number may be astronomical. Lemma 5 asserts that of all patterns of value 1 it is sufficient to consider only the very special class of elementary $x_1 \to x_n$ path flows. Of these, only those corresponding to unsaturated paths are relevant since, if P is saturated, relative to f_i, then $f_i + g_P$ is not feasible and hence not ideal. Of course the set of unsaturated paths may still be extremely large. Fortunately a fairly efficient constructive process can be described for determining one that minimizes $T(f_i + g_P)$.

The determination of the "best" $x_1 \to x_n$ path consists essentially of evaluating for each arc w of U and U' the increment of cost that would result if one unit of flow were added to f(w) and of selecting a path of arcs whose total incremental cost is minimum. Before making this precise it is necessary to introduce several new terms.

If $u \in U$, and $u \cong (x, y)$ let $\hat{u}$ denote the elementary $x \to y$ flow pattern obtained by considering u as an $x \to y$ path. Thus for all $v \in U$

$$\hat{u}(v) = \begin{cases} 1 & \text{if } v = u \\ 0 & \text{if } v \neq u \end{cases}$$

Similarly let $\hat{u}'$ denote the elementary $y \to x$ flow pattern corresponding to the $y \to x$ path u', so that for all $v \in U$

$$\hat{u}'(v) = \begin{cases} -1 & \text{if } v' = u' \\ 0 & \text{if } v' \neq u' \end{cases}$$

Thus with every $u \in U$ and every $u' \in U'$ is associated the flow pattern $\hat{u}$ or $\hat{u}'$, called the <u>characteristic flow</u> of arc u or u'. If P is a simple $x_1 \to x_n$ path the corresponding elementary flow pattern g_P can be written as $\Sigma_{w \in P} \hat{w}$.

If f is a feasible $x_1 \to x_n$ flow pattern, and $w \in U$ or $w \in U'$, the <u>effective cost of w, relative to f,</u> is denoted by e(w;f) and defined as follows:

$$e(w;f) = \begin{cases} T(f + \hat{w}) - T(f) & \text{if } (f + \hat{w}) \text{ is feasible} \\ \infty & \text{if } (f + \hat{w}) \text{ is infeasible} \end{cases}$$

If $w \cong (x, y)$, $e(w;f)$ may be considered as the increase in cost which results by taking a feasible pattern f and superimposing one unit of flow in w from x to y. In case this results in an infeasible pattern $f + \hat{w}$, the convention is adopted that an infinite increase in cost results.

If one considers the definition of the cost of flow in an arc in terms of a_1 and a_2 then $e(w;f)$ may be evaluated as follows:

For $u \in U$,

$$e(u;f) = \begin{cases} \infty & \text{if } f(u) = c(u) \\ a_1(u) & \text{if } 0 \leq f(u) < c(u) \\ -a_2(u) & \text{if } f(u) < 0 \end{cases}$$

For $u' \in U'$,

$$e(u';f) = \begin{cases} \infty & \text{if } f(u) = b(u) \\ a_2(u) & \text{if } b(u) < f(u) \leq 0 \\ -a_1(u) & \text{if } 0 < f(u) \end{cases}$$

If P is a simple $x_1 \to x_n$ path or a simple cycle in (X, W) the <u>effective cost of P</u>, relative to a feasible $x_1 \to x_n$ flow pattern f, is defined by

$$e(P;f) = \Sigma_{w \in P}\, e(w;f) \; .$$

One can establish the following result:

<u>Lemma 6</u>

If f is a feasible $x_1 \to x_n$ flow pattern in (X, U), P is a simple $x_1 \to x_n$ path or a simple cycle in (X, W), and g_P is the corresponding elementary path flow, then $e(P;f) = \infty$ if and only if P is saturated, i.e., if and only if $f + g_P$ is infeasible. If $e(P;f) < \infty$ so that $f + g_P$ is feasible then

$$T(f + g_P) = T(f) + e(P;f) \; .$$

Proof. If $e(P;f) = \infty$ then $e(w;f) = \infty$ for some $w \in P$. If $w = u \in U$ then $f(u) = c(u)$. If $w = u' \in U'$ then $f(u) = b(u)$. In the first case, u is saturated in the direction of its orientation. In the second case it is saturated in the reverse direction. In either case it is saturated in the direction associated with P, so P is saturated. Conversely if P is saturated there is an arc $w \in P$ whose flow cannot be feasibly increased by one unit in the direction of w. If $w = u \in U$ this implies that $f(u) = c(u)$. If $w = u' \in U'$ it implies that $f(u) = b(u)$. In either case $e(w;f) = \infty$; hence $e(P;f) = \infty$.

It remains to show that if $e(P;f) < \infty$ then $T(f + g_P) = T(f) + e(P;f)$. Let w be an arc of P, and assume first that $w = u \in U$. Let $T_u(f)$ denote the cost of flow in u relative to flow pattern f, and similarly for $T_u(f + g_P)$. If $f(u) \geq 0$ then $(f + g_P)(u) = f(u) + 1$, $T_u(f) = f(u)a_1$, and $T_u(f + g_P) = [f(u) + 1]a_1(u)$. Hence $T_u(f + g_P) - T_u(f) = a_1(u) = e(u;f)$. On the other hand if $f(u) < 0$ then $(f + g_P)(u) = f(u) + 1$, $T_u(f) = -f(u)a_2(u)$, and $T_u(f + g_P) = -[f(u) + 1]a_2(u)$. Hence $T_u(f + g_P) - T_u(f) = -a_2(u) = e(u;f)$. The reasoning is similar if $w = u' \in U'$.

Hence for every $u \in P$, $T_u(f + g_P) = T_u(f) + e(u;f)$, and for every $u' \in P$, $T_u(f + g_P) = T_u(f) + e(u';f)$. Adding the right-hand side of all these equations yields $T'(f) + e(P;f)$, where $T'(f)$ is the total cost of flow relative to f in all arcs involved in P. Adding the left-hand members yields $T'(f + g_P)$, where this expression has a similar meaning but relative to $f + g_P$. Since the cost of flow in arcs not associated with P is the same relative to f and to $f + g_P$ it follows that $T(f + g_P) = T(f) + e(P;f)$. This completes the proof.

It was seen earlier (Lemma 5) that if f is a nonmaximal ideal flow pattern then there exists a path P such that $f + g_P$ is ideal. In view of Lemma 6 one must clearly select P in such a way that e(P;f) is minimum.

Assume for the moment that an algorithm can be devised, which will be called Algorithm 1, that will perform the following function: given an ideal $x_1 \rightarrow x_n$ flow pattern f_i of value i, it will find a simple $x_1 \rightarrow x_n$ path P such that e(P;f) is minimum with respect to all such paths.

In terms of Algorithm 1, which will be described in detail shortly, one can state the following procedure that will always solve Problem A.

PROCEDURE FOR SOLVING PROBLEM A

(a) Let f_0 be the flow pattern that is defined by $f(u) = 0$ for all $u \in U$. This is an ideal $x_1 \rightarrow x_n$ flow pattern of value zero.

(b) In general, having determined an ideal flow pattern f_i of value i, apply Algorithm 1 and find a simple $x_1 \rightarrow x_n$ path P_i such that $e(P_i;f_i)$ is minimum.

(c) If $e(P_i;f_i) = \infty$ then $i = M$, i.e., there are no feasible $x_1 \rightarrow x_n$ flow patterns having larger values. So $\{f_0, f_1, \ldots, f_i\}$ is a complete solution to Problem A.

(c') If $e(P_i;f) < \infty$ then $f_{i+1} = f_i + g_i$ is an ideal $x_1 \rightarrow x_n$ flow pattern of value i + 1, where g_i is the elementary path flow corresponding to P_i. One can now replace f_i with f_{i+1} and repeat b.

In order to accelerate the process of finding the solution set $\{f_0, f_1, \ldots, f_M\}$ it is desirable to increase the value of successive patterns by more than one unit if possible. The following result provides a mechanism for doing so.

Lemma 7

Let f_i be an ideal, nonmaximal $x_1 \to x_n$ flow pattern in (X, U) having value i, and let P_i be a simple $x_1 \to x_n$ path in (X, W) that minimizes $e(P;f_i)$. For each arc $w \in P_i$ define Q_w as follows:

If $w \in U$,

$$Q_w = \begin{cases} c(w) - f(w) & \text{if } f(w) \geq 0 \\ -f(w) & \text{if } f(w) < 0 \end{cases}$$

If $w = u' \in U'$,

$$Q_w = \begin{cases} f(u) - b(u) & \text{if } f(u) \leq 0 \\ f(u) & \text{if } f(u) > 0 \end{cases}$$

Finally, let $Q = \min_{w \in P_i} Q_w$. Then for $1 \leq k \leq Q$,

$$f_{i+k} = f_i + kg_i$$

is an ideal $x_1 \to x_n$ flow pattern of value i+k, where g_i is the elementary path flow corresponding to P_i and $(kg_i)(u) = k \cdot g_i(u)$ for all $u \in U$.

Proof. In order to prove this lemma the following subsidiary result will first be established: If f_i, f_{i+1}, and f_{i+2} are ideal flow patterns having values i, i+1, and i+2 then

$$T(f_{i+2}) - T(f_{i+1}) \geq T(f_{i+1}) - T(f_i) .$$

According to Lemma 1, f_{i+2} can be expressed as

$$f_{i+2} = f_i + g_1 + g_2 + h ,$$

where g_1 and g_2 are elementary path flows, h is an appropriate zero valued flow (corresponding to a sum of cycle flows) and g_1, g_2, and h are conformal. Moreover from Lemma 2 it follows that $f_i + g_1 + g_2$ and $f_i + h$ are feasible. From Lemma 3 one obtains the relation

$$T(f_{i+2}) - T(f_i + g_1 + g_2) \geq T(f_i + h) - T(f_i) .$$

Since f_i is ideal the right-hand side is nonnegative. Hence

$$T(f_{i+2}) \geq T(f_i + g_1 + g_2) .$$

But since f_{i+2} is ideal, and $f_i + g_1 + g_2$ is also feasible and has value i+2, equality holds and $f_i + g_1 + g_2$ is also ideal. Applying Lemma 3 again yields the relation

$$T(f_i + g_1 + g_2) - T(f_i + g_1) \geq T(f_i + g_2) - T(f_i) .$$

Since f_{i+1} is ideal, $T(f_i + g_1) \geq T(f_{i+1})$ and $T(f_i + g_2) \geq T(f_{i+1})$. Hence

$$T(f_i + g_1 + g_2) - T(f_{i+1}) \geq T(f_{i+1}) - T(f_i) .$$

But the leftmost term is equal to $T(f_{i+2})$, and hence

$$T(f_{i+2}) - T(f_{i+1}) \geq T(f_{i+1}) - T(f_i) .$$

Now let f_i be an ideal flow pattern of value i, and again assume that patterns with values i+1 and i+2 exist; i.e., assume that $M \geq i+2$. Let R be a simple $x_1 \rightarrow x_n$ path minimizing $e(P;f_i)$, so that $f_i + g_R$ is ideal, where g_R is the corresponding elementary path flow.. Also, let S be a simple path that minimizes $e(P;f_i + g_R)$, so that $f_i + g_R + g_S$ is ideal. From the inequality established earlier in the proof $e(S;f_i + g_R) \geq e(R;f_i)$. Now if R had been such that $e(R;f_i + g_R) = e(R;f_i)$ then $f_i + 2g_R$ would necessarily also be ideal. Similarly if $e(R;f_i + 2g_R) = e(R;f_i)$ then

$f_i + 3g_R$ will be ideal. In general, $f_i + kg_R$ will be ideal so long as

$$e[R;f_i + (k-1)g_R] = e(R;f_i) .$$

Since the effective cost of a path is the sum of the effective costs of its component arcs, $f_i + kg_R$ will be ideal as long as

$$e[w;f_i + (k-1)g_R] = e(w;f_i)$$

for all $w \in R$. If $w \in U$ and $f(w) \geq 0$ each successive unit of flow will cost $a_1(w)$ until w is saturated, i.e., until flow equals $c(w)$. Then the effective cost changes to ∞. So the effective cost for arc w changes when $k = c(w) - f(w)$. If $w \in U$ and $f(w) < 0$ each unit of flow added in the direction associated with w will cancel a unit of flow, resulting in a reduction in cost of $-a_2(w)$. This will continue until all flow is canceled, at which time the effective cost of w changes from $-a_2(w)$ to $a_1(w)$. So in this case the effective cost changes when $k = -f(w)$. The two cases that arise when $w \in U'$ are very similar, and the detailed reasoning is omitted. In all cases the value of Q_w given in the statement of the lemma is the smallest value of k such that $e(w;f_i + kg_R) \neq e(w;f_i)$. So if $Q = \min_{w \in R} Q_w$ then Q is the largest integer k such that

$$T[f_i + kg_R] - T[f_i + (k-1)g_R] = T[f_i + g_R] - T[f_i] .$$

Hence $f_i + kg_R$ is ideal, for $k = 1, 2, \ldots, Q$. This completes the proof.

It follows from the fact that $e(P_i;f) < \infty$ that $k \geq 1$. In many applications k will be large if the $x_1 \rightarrow x_n$ capacity is large, and the number of repetitions of step b of the above procedure will be greatly reduced.

An algorithm for finding P_i that minimizes $e(P;f_i)$ is obtained by treating $e(w;f_i)$ as the distance from x to y, where $w \epsilon W$ and $w \cong (x, y)$. The problem of finding P_i is then one of finding the shortest distance from x_1 to x_n through the network (X, W). Efficient algorithms for finding shortest paths exist, provided the distances are such that no cycle has negative total length. Fortunately this is the case. The following result can be established:

Lemma 8

A necessary and sufficient condition that f be an ideal $x_1 \to x_n$ flow pattern in (X, U) is that $e(C;f) \geq 0$ for every simple cycle C in (X, W).

Proof. If f is ideal then certainly $e(C;f) \geq 0$ for all simple cycles C. For if $e(C;f) < 0$ for some C then from Lemma 6 it would follow that $T(f + g_C) < T(f)$, contradicting the fact that f is ideal.

Now suppose that $e(C;f) \geq 0$ for all simple cycles C. Let g be any $x_1 \to x_n$ flow pattern having the same value as f. Using Lemma 1, g can be written as $f + h_1 + \ldots + h_n$, where the h_i's are conformal cycle flows corresponding to appropriate cycles $C_1, C_2, \ldots, C_n$. Repeated application of Lemma 3 yields

$$T(f + h_1 + \ldots + h_n) - T(f + h_1 + \ldots + h_{n-1}) \geq T(f + h_n) - T(f)$$
$$T(f + h_1 + \ldots + h_{n-1}) - T(f + h_1 + \ldots + h_{n-2}) \geq T(f + h_{n-1}) - T(f)$$
$$\vdots$$
$$T(f + h_1 + h_2) - T(f + h_1) \geq T(f + h_2) - T(f)$$
$$T(f + h_1) - T(f) \geq 0 .$$

But the right-hand side of each inequality is ≥ 0 by assumption. Adding the left sides yields

$$T(f + h_1 + \ldots + h_n) - T(f) \geq 0 .$$

Since $f + h_1 + \ldots + h_n = g$ it follows that $T(g) \geq T(f)$. Since g was arbitrary this implies that f is ideal. The proof is complete.

In order to describe a specific algorithm having the characteristics of Algorithm 1 suppose that the arcs of W are arranged in a fixed sequence $w_1, w_2, \ldots, w_m$. (It is particularly useful to let u and u' be adjacent terms, since one uses the same basic data when considering u and u'.) For convenience let e_i denote $e(w_i;f)$ where f is the ideal flow pattern under consideration. The following procedure has the desired characteristics.

<u>Algorithm 1</u>

(a) Associate numbers, termed <u>labels</u>, with vertices, denoted by $V(x_i)$ as follows:

$$V(x_1) = 0$$
$$V(x_i) = \infty \text{ for } i = 2, 3, \ldots, n .$$

(b) Starting with w_1 take each arc in turn and do the following:

(i) If $w_i \cong (x_j, x_k)$ and $V(x_k) > V(x_j) + e_i$ replace $V(x_k)$ by the smaller quantity $V(x_j) + e_i$ and record w_i as the "approach arc" associated with vertex x_k. (If this is not the first iteration, w_i supersedes any approach arc previously associated with x_k.)

(ii) If $V(x_k) \leq V(x_j) + e_i$ do not modify $V(x_k)$, but consider arc w_{i+1} next.

When $i = m$ proceed to step c.

(c) If, in the course of considering $w_1, w_2, \ldots, w_m$ in step b, at least one label was modified, repeat step b.

(c') If no vertex was relabeled, stop. At this time $V(x_n)$ equals min $e(P;f)$ taken over all simple $x_1 \to x_n$ paths P. If $V(x_n) = \infty$, f is a maximum-valued flow. If $V(x_n) < \infty$ trace backward along the approach arc associated with x_n to its initial vertex x_p. Trace backward along x_p's approach arc to x_q and continue until x_1 is reached. This will yield a simple $x_1 \to x_n$ path (traced backward), minimizing $e(P;f)$.

The remainder of this section provides a proof of the fact that this algorithm will always produce the desired information in a finite number of steps regardless of the network (X, U) and the ideal flow f under consideration. Some of the properties of this algorithm will be needed in a more general form in connection with Problem C. For this reason the assumption that f is ideal is not made in establishing a number of intermediate results.

One "iteration" of the algorithm will mean one complete pass through the entire sets of arcs $\{w_1, w_2, \ldots, w_m\}$, relabeling a vertex whenever appropriate. From the time that a vertex other than x_1 first attains a finite label, it has associated with it an approach arc leading from another vertex that must also have a finite label. The approach arc associated with a vertex may change from time to time. Since the approach arc involved in assigning a new label supersedes the previous one (if any) identified with the vertex being relabeled, at no time does a vertex have more than one approach arc. Summarizing, if $V(x) = \infty$, x has no approach arc. If $V(x) < \infty$, x has precisely one approach arc, but it is not necessarily the same arc throughout the procedure.

Suppose that the algorithm is started and then interrupted at some time, and that $V(x) < \infty$ at this time, where $x \neq x_1$. Then x must have associated with it an approach arc, say $v_1 \cong (y_1, x)$. Similarly $V(y_1) < \infty$, so y_1 has an approach arc $v_2 \cong (y_2, y_1)$. Repeating this reasoning $v_3 \cong (y_3, y_2)$ is obtained, etc. In this process of tracing back along approach arcs one of three situations may arise. These are:

(a) Vertex x_1 may be reached via approach arcs $v_1, v_2, \ldots, v_m$ (i.e., x_1 is the initial vertex of v_m), and it may be impossible to trace back along another approach arc because x_1 has no associated approach arc. This will be the case if $V(x) = 0$, so that the original label of x_1 has not been altered. In this case the procedure has traced (backward) a simple $x_1 \rightarrow x_n$ path of approach arcs $\{v_m, v_{m-1}, \ldots, v_2, v_1\}$. (This path is necessarily simple because no vertex has two approach arcs leading to it.)

(b) The tracing procedure may lead to x_1, but x_1 may have been relabeled. In this case one can continue to trace backward, and must inevitably return to some vertex reached earlier since each step leads to another vertex with a finite label and every such vertex has an approach arc leading to still another vertex. Once a vertex is repeated in this tracing process, a cycle of approach arcs has been traversed (backward) and further tracing will merely duplicate the route already covered. In this case the set of approach arcs discovered by tracing takes the form of a cycle together with a path from the repeated vertex to x. (If the vertex repeated is x itself the set of arcs is simply a cycle.)

(c) The tracing procedure may repeat a vertex without ever reaching x_1. In this case the form of the set of approach arcs traced is the same as in situation b, being either a cycle together with a path from some vertex of the cycle to x, or else simply a cycle containing x. The only difference is that x_1 is not incident with any of the arcs traced.

In the remainder of this section for $x \neq x_1$, A(x) will denote the set of approach arcs traced backward from x, assuming that $V(x) < \infty$, i.e., that x has an approach arc. The set A(x) is called the <u>approach set associated with x</u>. It will be shown that if the flow pattern f under consideration is ideal, case a always holds, so that A(x) is always a simple $x_1 \rightarrow x$ path. If f is not ideal, sooner or later case b or case c arises for at least one vertex x. The important fact then is that there is a cycle of approach arcs and that such a cycle can be used to reduce the cost of f.

The symbol A will denote the total set of approach arcs associated with vertices at a given time during execution of the algorithm.

<u>Lemma 9</u>

If Algorithm 1 is applied to a feasible $x_1 \rightarrow x_n$ flow pattern and if the procedure is halted at any time then, if $w \in A$ at this time and if $w \cong (x, y)$,

$$V(y) \geq V(x) + e(w;f)$$

<u>Proof.</u> Note first that in this and in subsequent lemmas that investigate the status at an arbitrary time during the algorithm, the labels

and approach arcs referred to are those in effect at the time of interruption unless the contrary is specified.

Since w is the approach arc associated with y at this time it follows that $V(y) = V^*(x) + e(w;f)$ where $V^*(x)$ is the label that was associated with x at the time y attained its present label. But $V^*(x) \geq V(x)$ since the procedure never increases labels. It follows that $V(y) \geq V(x) + e(w;f)$, which completes the proof.

Lemma 10

If Algorithm 1 is applied to a feasible $x_1 \to x_n$ flow pattern and if P is any simple $x_1 \to y$ path having k arcs then after k or fewer iterations of the algorithm the following relation holds:

$$V(y) \leq e(P;f) .$$

Proof. Let $P = \{w_1, w_2, \ldots, w_k\}$ where $w_1 \cong (x_1, y_1)$, $w_2 \cong (y_1, y_2)$, $\ldots$, $w_k \cong (y_{k-1}, y)$. Let $V_t(z)$ denote the label of vertex z after t iterations, with the convention that $V_0(z)$ denotes the original label of z. Then clearly

$$V_1(y_1) \leq V_0(x_1) + e(w_1;f)$$

since every arc, including w_1, is considered as a candidate for relabeling y_1. Similarly,

$$V_2(y_2) \leq V_1(y_1) + e(w_2;f) \leq V_0(x_1) + e(w_1;f) + e(w_2;f) .$$

By repeating this reasoning this relation is ultimately obtained:

$$V_k(y) \leq V_0(x_1) + \Sigma_1^k e(w_i;f) = e(P;f) .$$

The second relation holds since $V_0(x_1) = 0$. If the algorithm terminates after fewer than k iterations then at the time of its termination $V(y_1) \leq V(x_1) + e(w_1;f), \ldots, V(y_k) \leq V(y_{k-1}) + e(w_k;f)$. Since $V(x_1) \leq 0$ this again implies that $V(y_k) \leq e(P;f)$. This completes the proof.

Lemma 11

If Algorithm 1 is applied to a feasible $x_1 \to x_n$ flow pattern f and if at some time the set of approach arcs A contains a cycle then f is not ideal.

Proof. Suppose that $C = \{w_1, w_2, \ldots, w_k\}$ is a set of approach arcs that forms a cycle. (Note that C must be a simple cycle because no two arcs of A ever terminate at the same vertex.) Assume that $w_1 \cong (y_1, y_2)$, $\ldots, w_k \cong (y_k, y_1)$. By applying Lemma 9 the following inequalities are obtained:

$$V(y_2) \geq V(y_1) + e(w_1;f)$$
$$V(y_3) \geq V(y_2) + e(w_2;f)$$
$$\vdots$$
$$V(y_1) \geq V(y_k) + e(w_k;f)\ .$$

It follows that $0 \geq \Sigma_1^k\, e(w_i;f) = e(C;f)$. Now let y_j be the first vertex of the set $\{y_1, y_2, \ldots, y_k\}$ which attained its present label. Then $V(y_j) = V^*(y_{j-1}) + e(w_j;f)$, where $V^*(y_{j-1})$ was the label of y_{j-1} at that time. By the way in which y_j was chosen $V(y_{j-1}) < V^*(y_{j-1})$. It follows that one of the above inequalities is strict, so that one obtains the sharper relation $0 > e(C;f)$. But then it follows from Lemma 8 that f cannot be ideal. This completes the proof.

Lemma 12

If Algorithm 1 is applied to a feasible $x_1 \rightarrow x_n$ flow pattern f and if $V(x_1) < 0$ at some stage then from this time on A contains a simple cycle.

Proof. Assume the algorithm is stopped some time after $V(x_1)$ becomes negative, with A denoting the set of approach arcs at the time of stopping. Let w_1 be the approach arc associated with x_1, and let y_1 be the vertex such that $w_1 \cong (y_1, x_1)$. Similarly let w_2 be the approach arc associated with y_1, and let y_2 be such that $w_2 \cong (y_2, y_1)$. One can continue to trace backward obtaining $\{w_3, w_4, \ldots\}$ and $\{y_3, y_4, \ldots\}$. Let k be the smallest integer such that $w_k \cong (y_k, y_{k-1})$, and such that y_k coincides with x_1 or with y_j for some $j < k$. (A repetition must occur eventually, since there are only a finite number of vertices.) If $y_k = x_1$ then $\{w_1, w_2, \ldots, w_k\}$ is a simple cycle. If $y_k = y_j$ then $\{w_{j+1}, w_{j+2}, \ldots, w_k\}$ is a simple cycle. In any case A contains a simple cycle. This completes the proof.

The following result is an immediate consequence of the last two lemmas:

Lemma 13

If Algorithm 1 is applied to a feasible $x_1 \rightarrow x_n$ flow pattern f and if at some time $V(x_1) < 0$ then f is not ideal.

Lemma 10 established the fact that $V(y)$ is less than or equal to the total effective cost $e(P;f)$ of any $x_1 \rightarrow y$ path P after a sufficient number of iterations. Lemma 14 establishes a reverse inequality. The two results together are then used to show that after a sufficient number of iterations $e(P;f)$ is precisely equal to $V(y)$, provided that f is ideal and that y ever attains a finite label.

Lemma 14

If Algorithm 1 is applied to a feasible $x_1 \to x_n$ flow pattern f and if the process is interrupted at any time then, if $A(y)$ is a simple $x_1 \to y$ path P for some vertex $y = x_1$, the inequality $V(y) \geq e(P;f)$ must hold.

Proof. Suppose $P = \{w_1, w_2, \ldots, w_k\}$ where $w_1 \cong (x_1, y_1)$, $w_2 \cong (y_1, y_2), \ldots, w_k \cong (y_{k-1}, y)$. It follows from Lemma 9 that

$$V(y_1) \geq V(x_1) + e(w_1;f)$$
$$V(y_2) \geq V(y_1) + e(w_2;f)$$
$$\vdots$$
$$V(y) \geq (V_{k-1}) + e(w_k;f) \ .$$

Summing, and using the fact that $V(x_1) = 0$ [because $A(y)$ is a simple path] $V(y) \geq e(P;f)$ is obtained. This completes the proof.

Theorem 15

If Algorithm 1 is applied to a nonmaximal ideal $x_1 \to x_n$ flow pattern in a Type A network having n vertices the algorithm will terminate (cease to reduce any labels) after n or fewer iterations or complete "passes" through the set of arcs. On termination, if $V(y) < \infty$, $A(y)$ will be a simple $x_1 \to y$ path that minimizes $e(P;f)$ over all such paths. In particular $A(x_n)$ will be an optimal $x_1 \to x_n$ path.

Proof. Since f is ideal, Lemma 11 states that at no time will the set of approach arcs contain a cycle. So for every $y \neq x_1$, either $V(y) = \infty$ or else $A(y)$ is a simple $x_1 \to y$ path P. In the latter case $V(y) \geq e(P;f)$ according to Lemma 14. So $V(y)$ is always as large as the effective cost of some simple $x_1 \to y$ path. But if P_y is an $x_1 \to y$ path that minimizes $e(P;f)$, and if P_y has k arcs then $V(y) \leq e(P_y;f)$ after k or fewer iterations, according to Lemma 10. Strict inequality cannot hold, so $V(y) = e(P_y;f)$ after k or fewer iterations. But $k \leq n$ because a simple path cannot have more than n arcs. So those vertices other than x_1 that ever have finite labels achieve their smallest labels after n or fewer iterations. As for x_1 itself Lemma 13 asserts that it is never relabeled. It remains to show that if $V(y) < \infty$ on termination then $A(y)$ is a simple $x_1 \to y$ path of minimum effective cost. [Note that P_y was any effectively "cheapest" path in the above argument. $A(y)$ may be a different one.] Let P_1 denote $A(y)$ on termination. Since $V(y) = e(P_y;f)$, and since $e(P_1;f) \geq e(P_y;f)$ because P_y is optimal it follows from Lemma 10 that $e(P_1;f) = e(P_y;f)$, so P_1 also minimizes $e(P;f)$. In particular if $y = x_n$, $V(x_n) < \infty$ on termination because f is not maximal, by assumption. So the above remarks apply, and $A(x_n)$ will be a simple $x_1 \to x_n$ path minimizing $e(P;f)$. This completes the proof.

If Algorithm 1 is applied to a maximal ideal $x_1 \to x_n$ flow pattern the only difference is that, on termination (which still must occur after n or fewer iterations), $V(x_n) = \infty$. Hence if the algorithm is applied to any ideal pattern the procedure will terminate after n or fewer iterations. If $V(x_n) = \infty$, f is maximal. If $V(x_n) < \infty$ then $A(x_n)$ is an $x_1 \to x_n$ path minimizing $e(P;f)$. Hence Algorithm 1 has the properties required for the procedure given earlier to solve Problem A. (Several of the lemmas used to establish this fact will also be used to devise a procedure for solving Problem C.)

SOLUTION OF PROBLEM B

This section describes a complete solution to Problem B. Recall that a Type B network is a weighted and capacitated network (X, U) such that $b(u) > 0$ for at least one arc $u \in U$. Moreover two vertices x_1 and x_n are designated as the source and sink of all flow patterns under consideration. The problem is that of finding an ideal $x_1 \to x_n$ flow pattern for each value realized by feasible $x_1 \to x_n$ patterns—with the possibility that there may be no such values in the given network.

In essence the procedure given here for solving Problem B consists of: (a) associating with the Type B network a related Type A network $(\overline{X}, \overline{U})$, (b) solving Problem A for the related network, and (c) interpreting the resulting ideal flow patterns as the solution to the Type B problem.

The construction of the associated network $(\overline{X}, \overline{U})$ employed here is a modification of a construction used by Berge,[2] and the procedure for finding a complete set of ideal Type B flows is an extension of a procedure, indicated in that reference, for determining a feasible Type B flow pattern of least cost.

Given a Type B network (X, U); construct a Type A network $(\overline{X}, \overline{U})$ as follows:

(a) Augment X by adding two vertices $\overline{x}_1$ and $\overline{x}_n$. (This notation is adopted because these vertices will constitute the source and sink of flow patterns in the augmented network.) Let $\overline{X}$ denote X together with $\overline{x}_1$ and $\overline{x}_2$.

(b) Let S denote the set of arcs of U such that $b(u) > 0$. (The set S occurs frequently in this section and always has the above meaning.) If $u \epsilon S$ and $u \cong (x, y)$, construct two additional arcs u° and $u^{\circ\circ}$ where $u^{\circ} \cong (\overline{x}_1, y)$ and $u^{\circ\circ} \cong (x, \overline{x}_n)$. Let U° and $U^{\circ\circ}$ denote the set of arcs of type u° and $u^{\circ\circ}$ respectively. Each of these sets is in one-to-one correspondence with S.

(c) Construct two additional arcs w° and $w^{\circ\circ}$, where $w^{\circ} \cong (\overline{x}_1, x_1)$ and $w^{\circ\circ} \cong (x_n, \overline{x}_n)$. Let $\overline{U}$ denote the totality of arcs of U, U°, and $U^{\circ\circ}$, plus the two arcs w° and $w^{\circ\circ}$. (Figure 5 is an example intended to clarify the above construction. In this example the heavy-line arcs constitute the set S.)

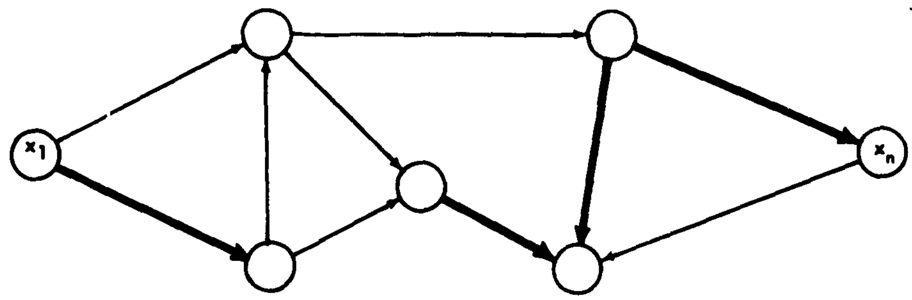
(d) Lower and upper arc bounds denoted by $\overline{b}(u)$ and $\overline{c}(u)$ for $u \epsilon \overline{U}$ are defined as follows:

$$\overline{b}(u) = b(u) \text{ and } \overline{c}(u) = c(u) \text{ if } u \epsilon U\text{-}S.^*$$

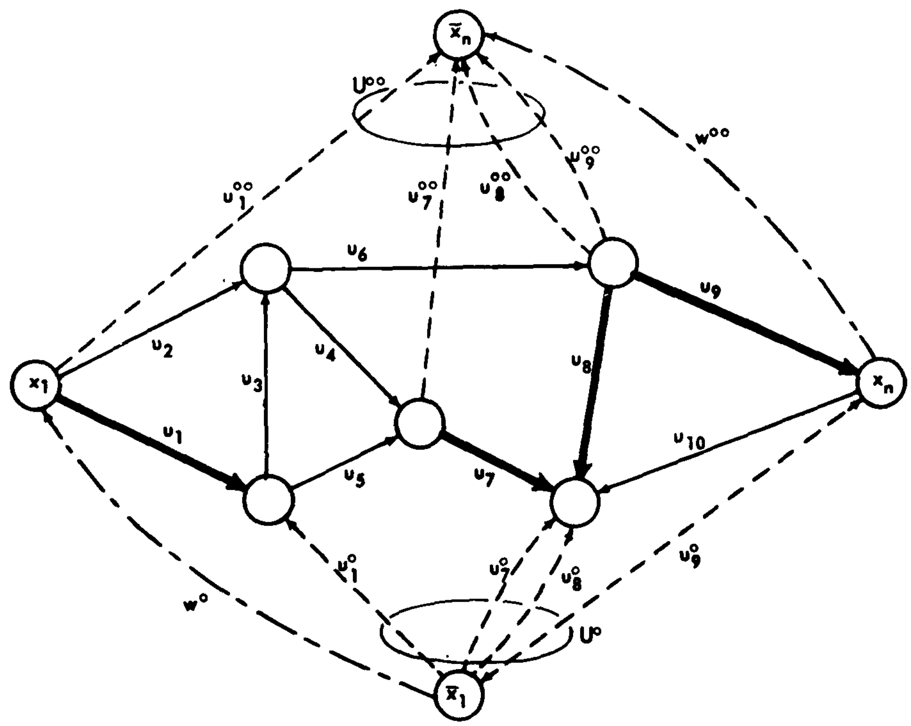
$$\overline{b}(u) = 0 \text{ and } \overline{c}(u) = c(u) - b(u) \text{ if } u \epsilon S.$$

$$\overline{b}(u^{\circ}) = 0 \text{ and } \overline{c}(u^{\circ}) = b(u) \text{ for } u^{\circ} \epsilon U^{\circ}.$$

*U-S denotes the arcs of U that are not in S, i.e., those $u \epsilon U$ such that $b(u) \leq 0$.



a. Given network (X, U)†



b. Associated network $(\bar{X}, \bar{U})$†

Fig. 5—Relation between Networks (X, U) and $(\bar{X}, \bar{U})$

†Bold arcs indicate the set S.

$\bar{b}(u^{\circ\circ}) = 0$ and $\bar{c}(u^{\circ\circ}) = b(u)$ for $u^{\circ\circ} \in U^{\circ\circ}$.

$\bar{b}(w^{\circ}) = \bar{b}(w^{\circ\circ}) = 0$

$\bar{c}(w^{\circ}) = \bar{c}(w^{\circ\circ}) = K_1$.

Here K_1 is any integer known to exceed the value of every feasible $x_1 \rightarrow x_n$ flow pattern in (X, U).

(e) Unit costs denoted by $\bar{a}_1(u)$ and $\bar{a}_2(u)$ are defined as follows:

$\bar{a}_1(u) = a_1(u)$ and $\bar{a}_2(u) = a_2(u)$ for $u \in U$.

$\bar{a}_1(u^{\circ}) = \bar{a}_2(u^{\circ}) = 0$ for $u^{\circ} \in U^{\circ}$.

$\bar{a}_1(u^{\circ\circ}) = \bar{a}_2(u^{\circ\circ}) = 0$ for $u^{\circ\circ} \in U^{\circ\circ}$.

$\bar{a}_1(w^{\circ}) = \bar{a}_1(w^{\circ\circ}) = K_2$.

$\bar{a}_2(w^{\circ}) = \bar{a}_2(w^{\circ\circ}) = 0$.

Here K_2 is an integer that exceeds $\Sigma_U \max\{a_1(u), a_2(u)\}$. (The reasons for specifying K_1 and K_2 in this manner will be clarified later.)

If $\bar{x}_1$ and $\bar{x}_n$ are designated as source and sink in the augmented network then $(\bar{X}, \bar{U})$ together with functions $\bar{b}$, $\bar{c}$, $\bar{a}_1$, and $\bar{a}_2$ define a Type A network. Whenever feasible or ideal flow patterns in $(\bar{X}, \bar{U})$ are mentioned these are intended to be $\bar{x}_1 \rightarrow \bar{x}_n$ patterns that are feasible or ideal relative to $\bar{b}$, $\bar{c}$, $\bar{a}_1$, and $\bar{a}_2$. Patterns in (X, U) are always related to b, c, a_1, and a_2. Flow patterns in $(\bar{X}, \bar{U})$ will be denoted by symbols such as $\bar{f}$ and $\bar{g}$ to further distinguish them from flows in (X, U).

Let $\bar{f}$ be a feasible flow pattern in $(\bar{X}, \bar{U})$ which saturates U° and $U^{\circ\circ}$, i.e., one such that $\bar{f}(u^{\circ}) = \bar{c}(u^{\circ})$ for all $u^{\circ} \in U^{\circ}$ and $\bar{f}(u^{\circ\circ}) = \bar{c}(u^{\circ\circ})$ for all $u^{\circ\circ} \in U^{\circ\circ}$. Such patterns, and only such, are called <u>transformable flow patterns</u> in (X, U). It will be shown that transformable flow patterns in $(\overline{X}, \overline{U})$ correspond to feasible patterns in (X, U) and that ideal transformable patterns in $(\overline{X}, \overline{U})$ correspond to ideal patterns in (X, U).

Let $\overline{f}$ be an arbitrary transformable flow pattern in $(\overline{X}, \overline{U})$. A transformation ϕ will now be introduced that maps $\overline{f}$ into a flow pattern f in (X, U). Specifically, for every $u \in U$ define $f(u)$ as follows:

$$f(u) = \begin{cases} \overline{f}(u) & \text{if } u \in U-S \\ \overline{f}(u) + b(u) & \text{if } u \in S \end{cases}$$

The pattern f related to $\overline{f}$ in this way will be denoted sometimes by $\phi\overline{f}$. Note that if $\overline{f}_1$ and $\overline{f}_2$ are different transformable flow patterns in $(\overline{X}, \overline{U})$ then $\phi\overline{f}_1$ and $\phi\overline{f}_2$ will be different functions on U. For if $\overline{f}_1$ and $\overline{f}_2$ coincide on U they will coincide also on U°, $U^{\circ\circ}$, w°, and $w^{\circ\circ}$. (All transformable flow patterns have the same values on arcs of U° and $U^{\circ\circ}$, so that it is only necessary to consider w° and $w^{\circ\circ}$. But $\overline{f}_1$ and $\overline{f}_2$ must be such that input equals output at x_1 and x_n. Since the values of arc flow are the same for all arcs incident with x_1, except possibly w°, they must agree also on w°. A similar consideration applies for $w^{\circ\circ}$.) Thus if $\overline{f}_1$ and $\overline{f}_2$ are distinct they must differ on an arc of U, so that $f_1 = \phi\overline{f}_1$ will not coincide with $f_2 = \phi\overline{f}_2$. Expressed differently, ϕ is a one-to-one mapping of the set of transformable $\overline{x}_1 \to \overline{x}_n$ flow patterns in $(\overline{X}, \overline{U})$ onto a certain set of flow patterns in (X, U). This latter set is in fact the set of all feasible $x_1 \to x_n$ flow patterns in (X, U), as will be seen presently.

<u>Lemma 16</u>

If $\overline{f}$ is a transformable $\overline{x}_1 \to \overline{x}_n$ flow pattern of value k in $(\overline{X}, \overline{U})$ then $f = \phi\overline{f}$ is a feasible $x_1 \to x_n$ flow pattern of value $k - \Sigma_S b(u)$ in (X, U).

Proof. Note first that each arc flow is feasible. For if $u \in U-S$, $f(u) = \bar{f}(u)$, $b(u) = \bar{b}(u)$ and $c(u) = \bar{c}(u)$, so that the feasibility of $f(u)$ follows from that of $f(u)$. If $u \in S$ then $f(u) = \bar{f}(u) + b(u)$, $\bar{b}(u) = 0$, and $\bar{c}(u) = c(u) - b(u)$. Since $\bar{f}$ is feasible $0 \leq \bar{f}(u) \leq \bar{c}(u)$. Consequently $b(u) \leq f(u) \leq c(u)$, so $f(u)$ is feasible.

It will be shown next that if $x \in X$ and x is neither x_1 nor x_n the net output at x is zero. The summation $\Sigma_{U(\rightarrow x)} f(u)$ can be written as $\Sigma_1 \bar{f}(u) + \Sigma_2 \bar{f}(u)$ where Σ_1 ranges over arcs of $U \cap \bar{U}(\rightarrow x)$* and Σ_2 ranges over $U^\circ \cap \bar{U}(\rightarrow x)$. For every $u \in S \cap \bar{U}(\rightarrow x)$, Σ_1 contains a term $\bar{f}(u)$, and Σ_2 contains a term $\bar{c}(u^\circ) = b(u)$ which together equal $\bar{f}(u) + b(u) = f(u)$. For every $u \in (U-S) \cap \bar{U}(\rightarrow x)$, Σ_1 contains a term $\bar{f}(u)$, and Σ_2 has no corresponding term, since $b(u) \leq 0$ for arcs of U-S and no corresponding arc u° exists. Also, $\bar{f}(u) = f(u)$ in this case. It follows that $\Sigma_{U(\rightarrow s)} f(u) = \Sigma_{\bar{U}(\rightarrow x)} \bar{f}(u)$ since each term of the first summation can be paired with one or two terms of the second summation, depending on whether $u \in (U-S)$ or whether $u \in S$. A very similar argument shows that $\Sigma_{U(x \rightarrow)} f(u) = \Sigma_{\bar{U}(x \rightarrow)} \bar{f}(u)$. Since the net output of $\bar{f}$ at x is zero it follows that the net output of f at x is zero. Thus f is either a feasible $x_1 \rightarrow x_n$ flow pattern or a feasible $x_n \rightarrow x_1$ flow pattern in (X, U). It will next be shown that the former possibility is always true. (The latter will also be true whenever the value of f is zero.)

Consider now vertex $x = x_1$. One can write $\Sigma_{\bar{U}(\rightarrow x)} \bar{f}(u) = \Sigma_1 \bar{f}(u) + \Sigma_2 \bar{f}(u) + \bar{f}(w^\circ)$ where Σ_1 and Σ_2 have the same meanings as before. By matching

*$A \cap B$ denotes the set of elements common to sets A and B.

all terms of Σ_1 and Σ_2 with terms of $\Sigma_{U(\rightarrow x)} f(u)$, only the arc flow $\overline{f}(w^\circ)$ has no counterpart in f. So $\Sigma_{U(\rightarrow x)} f(u) = \Sigma_{\overline{U}(\rightarrow x)} \overline{f}(u) - \overline{f}(w^\circ)$. As before $\Sigma_{U(x\rightarrow)} f(u) = \Sigma_{\overline{U}(x\rightarrow)} \overline{f}(u)$. Since net output of $\overline{f}$ at x_1 is zero it follows that the net output of f at x_1 is $\overline{f}(w^\circ)$. So f is a feasible $x_1 \rightarrow x_n$ flow pattern of value $\overline{f}(w^\circ)$ in (X, U). Now the value of $\overline{f}$ is $k = \Sigma_{U^\circ} \overline{f}(u) + \overline{f}(w^\circ)$ and $\Sigma_{U^\circ} \overline{f}(u) = \Sigma_S b(u)$. Hence the value of f is $k - \Sigma_S b(u)$ as asserted. This completes the proof.

It was noted above that no flow pattern in (X, U) is the transform of more than one transformable flow pattern in $(\overline{X}, \overline{U})$. On the other hand it can readily be shown that an arbitrary feasible $x_1 \rightarrow x_n$ flow pattern f in (X, U) is the transform of some transformable flow pattern in $(\overline{X}, \overline{U})$. To show this, define a transformation Ψ as follows: $\Psi f = \overline{f}$, where $\overline{f}$ is the flow pattern in $(\overline{X}, \overline{U})$ defined by:

$$\begin{aligned}
\overline{f}(u) &= f(u) \text{ if } u \in U\text{-}S. \\
\overline{f}(u) &= f(u) - b(u) \text{ if } u \in S. \\
\overline{f}(u^\circ) &= b(u) \text{ if } u^\circ \in U^\circ. \\
\overline{f}(u^{\circ\circ}) &= b(u) \text{ if } u^{\circ\circ} \in U^{\circ\circ}. \\
\overline{f}(w^\circ) &= \overline{f}(w^{\circ\circ}) = k, \text{ where k is the value of f.}
\end{aligned}$$

It is clear that $\overline{f}$ is a feasible $\overline{x}_1 \rightarrow \overline{x}_n$ flow pattern in $(\overline{X}, \overline{U})$. (The fact that net output is zero for all $x \in X$ is established by introducing Σ_1 and Σ_2 as before and applying essentially the same argument as before.) The value of $\overline{f}$ is $k + \Sigma_S b(u)$ where k is the value of f.

The transformation Ψ is the inverse of ϕ, satisfying $\Psi(\phi \overline{f}) = \Psi f = \overline{f}$ and $\phi(\Psi f) = \phi \overline{f} = f$. So Ψ will be denoted by ϕ^{-1}. Summarizing the above observations gives Theorem 17.

Theorem 17

ϕ is a one-to-one mapping of the set of all transformable $\bar{x}_1 \to \bar{x}_n$ flow patterns in $(\bar{X}, \bar{U})$ onto the set of all feasible $x_1 \to x_n$ flow patterns in (X, U). If $\bar{f}$ is transformable and has value k then $\phi\bar{f}$ has value $k - \Sigma_S b(u)$. The transformation Ψ defined above is the inverse mapping ϕ^{-1} of all feasible $x_1 \to x_n$ flows onto all transformable $\bar{x}_1 \to \bar{x}_n$ flows.

The costs as well as the values of a transformable $\bar{x}_1 \to \bar{x}_n$ flow pattern $\bar{f}$ and its image $f = \phi\bar{f}$ are connected by a simple relation. The following result establishes this relation.

Theorem 18

If $f = \phi\bar{f}$ then

$$T(f) = T(\bar{f}) - 2\bar{a}_1(w^{*})\bar{f}(w^{*}) + \Sigma_S a_1(u)b(u) .$$

Since $\bar{f}(w^{*})$ was seen earlier to be the value of f, and $\Sigma_S a_1(u)b(u)$ is a constant, this means that T(f) differs from $T(\bar{f})$ by an additive constant and a term proportional to the value of f.

Proof. For $u \in U-S$, $\bar{f}(u) = f(u)$ and $\bar{a}_i(u) = a_i(u)$ for $i = 1$ and 2, so the costs of f and $\bar{f}$ are the same on U-S. For $u \in S$ the cost of f is $\Sigma_S a_1(u)f(u)$ and that of $\bar{f}$ is $\Sigma_S a_1(u)\,[f(u) - b(u)]$, so that the cost of f exceeds that of $\bar{f}$ by $\Sigma_S a_1(u)b(u)$ on S. Since f is only defined on arcs of S and U-S the total cost of f is accounted for above. Since $a_1(u) = a_2(u) = 0$ for $u \in U^{*}$ or $u \in U^{**}$ the only additional cost of $\bar{f}$ is associated with arc flows $\bar{f}(w^{*})$ and $\bar{f}(w^{**})$. But since $\bar{f}(w^{*}) = \bar{f}(w^{**}) \geq 0$ for transformable flows, and $\bar{a}_1(w^{*}) = \bar{a}_1(w^{**})$ by definition, the total cost in w^{*} and w^{**} is $2\bar{a}_1(w^{*})\bar{f}(w^{*})$. The formula given in the statement of this theorem is simply a consolidation of the above relations.

As a corollary to the preceding theorem it may be noted that if f is a feasible $x_1 \to x_n$ flow pattern of value k, and $\bar{f} = \phi^{-1}f$, then

$$T(f) = T(\bar{f}) - 2k\bar{a}_1(w^*) + \Sigma_S a_1(u)b(u) ,$$

so that the costs of f and $\bar{f}$ differ by a constant if the value of f is fixed. Therefore to find a minimum-cost (i.e., ideal) $x_1 \to x_n$ flow pattern of value k it suffices to find a minimum-cost transformable $\bar{x}_1 \to \bar{x}_n$ flow pattern $\bar{f}$ such that the value of $\phi\bar{f}$ is k. Since the value of $\bar{f}$ exceeds that of $f = \phi\bar{f}$ by $\Sigma_S b(u)$ it suffices to find a transformable $\bar{f}$ whose value is $k + \Sigma_S b(u)$ and such that no other transformable flow having this value has smaller cost.

Nearly all the information required for a general solution to Problem B is now at hand. The remaining gap to be filled is to show that every <u>non-transformable</u> $\bar{x}_1 \to \bar{x}_n$ flow pattern of a given value costs more than a minimum-cost transformable $\bar{x}_1 \to x_n$ flow pattern, provided that any transformable patterns having this value exist. For otherwise, if one simply computes ideal $\bar{x}_1 \to \bar{x}_n$ flows for all feasible values, these flows may not be transformable and hence will not yield corresponding ideal $x_1 \to x_n$ flows. The next result removes this danger. In the course of doing so, the reason for associating a large unit cost (viz., K_2) with w^* and w^{**} will be clarified.

<u>Lemma 19</u>

If any transformable $\bar{x}_1 \to \bar{x}_n$ flow patterns of value k exist then every ideal pattern of value k is transformable.

Proof. Assume that $\overline{f}_1$ and $\overline{f}_2$ are feasible $\overline{x}_1 \rightarrow \overline{x}_n$ flow patterns having the same value k, and that $\overline{f}_1$ is transformable whereas $\overline{f}_2$ is not. It will be shown that $\overline{f}_2$ cannot be ideal. Consider the zero-valued flow pattern $\overline{f}_1 - \overline{f}_2$. By Lemma 1 it is possible to write $\overline{f}_1 - \overline{f}_2 = g_1 + g_2 + \ldots + g_n$ where the g_i's are appropriate conformal elementary cycle flows. Since $\overline{f}_2$ is not transformable U° and $U^{\circ\circ}$ are not both saturated by $\overline{f}_2$, whereas $\overline{f}_1$ does saturate both U° and $U^{\circ\circ}$. Suppose $\overline{f}_2$ fails to saturate U°. Then $\overline{f}_1 - \overline{f}_2$ involves positive flow in one or more arcs $u^{\circ} \in U^{\circ}$, zero flow in the remaining arcs of U°, and negative flow (i.e., flow toward $\overline{x}_1$) in w°. At least one of the elementary cycle flows, say g_i, must be such that $g_i(u^{\circ}) = 1$ for some $u^{\circ} \in U^{\circ}$ and $g_i(w^{\circ}) = -1$, since the sum of all of the cycles accounts for all flow of $\overline{f}_1 - \overline{f}_2$. From Lemma 2 it follows that $\overline{f}_2 + g_i$ is a feasible flow pattern whose value is k. It will now be shown that $T(\overline{f}_2 + g_i) < T(\overline{f}_2)$, so that $\overline{f}_2$ cannot be ideal. Now $e(g_i;\overline{f}_2)$ is the sum of terms each of which is either plus or minus the unit cost of an arc of the elementary cycle associated with g_i. In particular one of the terms is $-K_2$ where K_2 was the unit cost of w°, i.e., $a_1(w^{\circ})$. If the cycle also contains arc $w^{\circ\circ}$, $g_i(w^{\circ\circ}) = -1$ for the same reasoning as that applied to w°. So $e(g_i;\overline{f}_2) = -K_2 + Q$ or $-2K_2 + Q$ where Q is the sum of the effective lengths of arcs of U involved in the cycle. (The cycle also contains an arc of type u° and possibly one of type $u^{\circ\circ}$, but since these have zero unit costs their effective lengths are zero and can be disregarded.) But it is clear that $K_2 > Q$ since K_2 was taken to be an integer greater than $\Sigma_U \max\{a_1(u), a_2(u)\}$ and hence is surely greater than the sum of the effective lengths relative to $\overline{f}_2$ of selected arcs of U. It follows that

$e(g_i;\bar{f}_2) < 0$ so that $T(\bar{f}_2 + g_i) < T(\bar{f}_2)$ by Lemma 6. Hence $\bar{f}_2$ is not ideal. (If $U^{\bullet}$ were saturated by $\bar{f}_2$ but $U^{\bullet\bullet}$ were not, the proof would be essentially the same.)

All the necessary facts that lead up to the following theorem, whose validity forms the basis for a general procedure for solving Problem B, have now been established.

Theorem 20

Let $\overline{M}$ denote the $\bar{x}_1 \rightarrow \bar{x}_n$ capacity of $(\overline{X}, \overline{U})$. Either there are no transformable $\bar{x}_1 \rightarrow \bar{x}_n$ flow patterns, or else there is an integer $\overline{m}$, where $0 \leq \overline{m} \leq \overline{M}$, such that there are no transformable $\bar{x}_1 \rightarrow \bar{x}_n$ flow patterns of value p for $0 \leq p < \overline{m}$, and every ideal pattern of value q, where $\overline{m} \leq q \leq \overline{M}$, is transformable.

Proof. Suppose there exists a transformable $\bar{x}_1 \rightarrow \bar{x}_n$ flow pattern of value i, where $i < \overline{M}$. It follows from Lemma 19 that any ideal $\bar{x}_1 \rightarrow \bar{x}_n$ flow pattern of value i is necessarily transformable. Let $\bar{f}_i$ be such a pattern. Since $\bar{f}_i$ is ideal but not maximal it follows from Lemma 5 that an ideal flow pattern $\bar{f}_{i+1}$ of value $i+1$ can be obtained by adding an appropriate elementary path flow to $\bar{f}_i$. But since every arc of type $u^{\bullet}$ and $u^{\bullet\bullet}$ is already saturated the first and last arcs of an unsaturated elementary path are necessarily $w^{\bullet}$ and $w^{\bullet\bullet}$. Hence $\bar{f}_{i+1}$ must be such that the flow in arcs $w^{\bullet}$ and $w^{\bullet\bullet}$ is increased by one, and arcs of types $u^{\bullet}$ and $u^{\bullet\bullet}$ are still saturated. Thus $\bar{f}_{i+1}$ is also transformable. If $\bar{f}_{i+1}$ is not maximal the argument may be repeated, so that ultimately ideal flow patterns of values $i, i+1, \ldots, \overline{M}$, which are all transformable, can be generated.

From Lemma 19 it follows that no feasible but nontransformable flow patterns of values i, i+1, . . ., $\overline{M}$ can be ideal. The proof is complete if $\overline{m}$ is taken as the smallest integer i such that a transformable flow pattern of value i exists.

Corollary

Since there is a feasible $x_1 \to x_n$ flow pattern of value k in (X, U) if and only if there is a transformable $\overline{x}_1 \to \overline{x}_n$ flow pattern of value $k + \Sigma_S b(u)$ in $(\overline{X}, \overline{U})$, it follows that there are feasible $x_1 \to x_n$ flow patterns of value k for $m \leq k \leq M$, where $m = \overline{m} - \Sigma_S b(u)$ and $M = \overline{M} - \Sigma_S b(u)$, and for no other values.

PROCEDURE FOR SOLVING PROBLEM B

Every Type B problem can be solved by the following method:

(a) Given a Type B network (X, U); construct the associated Type A network $(\overline{X}, \overline{U})$.

(b) Solve the Type A problem for this network, obtaining a sequence $\{\overline{f}_1, \overline{f}_2, \ldots, \overline{f}_{\overline{M}}\}$ where each $\overline{f}_i$ is an ideal $\overline{x}_1 \to \overline{x}_n$ flow pattern of value i in $(\overline{X}, \overline{U})$ and $\overline{M}$ is the $\overline{x}_1 \to \overline{x}_n$ capacity of $(\overline{X}, \overline{U})$.

(c) If none of the $\overline{f}_i$'s are transformable there are no feasible flows in (X, U).

(d) If at least one $\overline{f}_i$ is transformable then for some m it will be true that $\{\overline{f}_1, \ldots, \overline{f}_{\overline{m}-1}\}$ are nontransformable and $\{\overline{f}_{\overline{m}}, \ldots, \overline{f}_{\overline{M}}\}$ are transformable. It can then be asserted that $m = \overline{m} - \Sigma_S b(u)$ and $M = \overline{M} - \Sigma_S b(u)$ are the smallest and largest values for which feasible

$x_1 \to x_n$ flow patterns in (X, U) exist. Moreover $f_i = \phi\overline{f}_i + \Sigma_S b(u)$ is an ideal flow pattern of value i for $i = m, m+1, \ldots, M$.

This procedure, incorporating in step b the algorithm of the previous section, or any other known technique for solving Problem A, constitutes the complete solution to Problem B.

SOLUTION OF PROBLEM C

Let (X, U) be a weighted, capacitated network, and let f be a feasible $x_1 \rightarrow x_n$ flow pattern in (X, U). Suppose that one wishes to determine whether or not f is ideal. If the network is a Type A network, a theoretical criterion has already been given: f is ideal if and only if $e(C;f) \geq 0$ for every elementary cycle C in (X, W) where W denotes $U \cup U'$ as usual. Suppose f is not ideal. Since it has been established that

$$T(f + g_c) = T(f) + e(C;f) ,$$

where g_c is the elementary cycle flow associated with C, $f_1 = f + g_c$ is a feasible flow pattern having the same value as f but reduced cost, if $e(C;f) < 0$. The reasoning can be repeated, using f_1 in place of f. Either $e(C;f_1) \geq 0$ for all C or else one can find C such that $f_2 = f_1 + g_c$ has still smaller total cost. Proceeding in this fashion an f_p is ultimately obtained such that $e(C;f_p) \geq 0$ for all C. Then f_p must be ideal. An algorithm that locates a simple cycle C such that $e(C;f) < 0$ whenever such a cycle exists is described in this section. This algorithm is essentially the same as Algorithm 1, which was used earlier to solve Problem A.

Although Problem C is posed for networks of Type B as well as for those of Type A it is sufficient to produce a general solution for Type A networks. For if (X, U) is a Type B network let $(\overline{X}, \overline{U})$ denote the associated Type A network defined in the previous section. Using the notation of that section, if f is a feasible $x_1 \to x_n$ flow pattern in (X, U) then $\overline{f} = \phi^{-1} f$ is a transformable $\overline{x}_1 \to \overline{x}_n$ flow pattern in $(\overline{X}, \overline{U})$. Moreover f is ideal in (X, U) if and only if $\overline{f}$ is ideal in $(\overline{X}, \overline{U})$. Hence $\overline{f}$ can be tested to determine whether or not it is ideal. If it is not, suppose that it is changed into an ideal transformable pattern $\overline{f}_1$ having the same value. Then $f_1 = \phi \overline{f}_1$ is an ideal pattern in (X, U), having the same value as f. Hence Problem C in a Type B network can always be solved by solving a related problem in the associated Type A network.

For Type A networks, the key to solving Problem C is the following:

<u>Theorem 21</u>

If Algorithm 1 is applied to a feasible, nonideal $x_1 \to x_n$ flow pattern f and if every vertex eventually has a finite label then the algorithm will continue to relabel vertices after n iterations. If y is relabeled during the m^{th} iteration, where $m > n$, and if the procedure is interrupted immediately after this relabeling, then the approach set $A(y)$ will contain a cycle.

<u>Proof</u>. The second part of the theorem will be proved first. Suppose that y is relabeled on the m^{th} iteration, where $m > n$, and the algorithm is then halted. If $A(y)$ were a simple $x_1 \to y$ path P at this time it would follow from Lemma 14 that $V(y) \geq e(P;f)$. But P can have at most $(n - 1)$ arcs,

and hence it follows from Lemma 10 that $V(y) \leq e(P;f)$ before the m^{th} iteration started, which contradicts the fact that it was relabeled on the m^{th} iteration. Hence $A(y)$ cannot be a simple $x_1 \to y$ path. It was seen earlier that if $A(y)$ was not a path then it contained a cycle. So the second half of the theorem has been established.

It remains to show that some vertices will in fact be relabeled on iterations after the n^{th}. Since f is not ideal there is a simple cycle C such that $e(C;f) < 0$. This follows from Lemma 8. Suppose this cycle is $\{w_1, w_2, \ldots, w_k\}$, where $w_1 \cong (y_1, y_2)$, $w_2 \cong (y_2, y_3)$, . . ., $w_k \cong (y_k, y_1)$. Consider any time during the algorithm at which all vertices of the set $\{y_1, y_2, \ldots, y_k\}$ have finite labels. It has been assumed that from some point in time onward this will be the case. Suppose that at such a time the algorithm is halted. The following inequalities cannot all be true:

$$
\begin{aligned}
V(y_2) &\leq V(y_1) + e(w_1;f) \\
V(y_3) &\leq V(y_2) + e(w_2;f) \\
&\vdots \\
V(y_k) &\leq V(y_{k-1}) + e(w_{k-1};f) \\
V(y_1) &\leq V(y_k) + e(w_k;f)
\end{aligned}
$$

for this would imply that $0 \leq e(C;f)$. So at least one label can be improved. But this will always be the case, since the algorithm was halted at an arbitrary time after all vertices of the set $\{y_1, y_2, \ldots, y_k\}$ attained finite labels. So the algorithm will never arrive at a " stable" set of labels such that $V(y) \geq V(x) + e(w)$ for every $w \in W$ where $w \cong (x, y)$. This completes the proof.

Let f be a feasible $x_1 \to x_n$ flow pattern in a Type A network (X, U), with U' and W defined as usual. A new network $(X, \tilde{U})$ will now be introduced by adding additional arcs to U. Specifically, for every $x_i \neq x_1$ an arc v_i is created, with $v_i \cong (x_1, x_i)$. Flow bounds and costs are assigned as follows:

$$
\begin{aligned}
b(v_i) &= 0 \\
c(v_i) &= 1 \\
a_1(v_i) &= K \\
a_2(v_i) &= 0 \,.
\end{aligned}
$$

Here K is chosen to be an integer larger than $\Sigma_U \max [a_1(u), a_2(u)]$. Thus K is larger than the effective cost (relative to f) of any simple path joining two vertices by means of arcs of W, provided that none of the arcs is saturated.

If $\tilde{U}$ denotes U augmented by $\{v_2, v_3, \ldots, v_n\}$ and if $b(v_i)$, $c(v_i)$, $a_1(v_i)$, and $a_2(v_i)$ are defined in the above manner then $(X, \tilde{U})$ is again a Type A network. Moreover if we extend the definition of f by setting $f(u) = 0$ for $u = v_i$ and $i = 2, 3, \ldots, n$ then the extended function $\tilde{f}$ is a feasible $x_1 \to x_n$ flow pattern in $(X, \tilde{U})$. Ideal patterns in (X, U) and $(X, \tilde{U})$ are related by the following result:

<u>Lemma 22</u>

If f is a feasible $x_1 \to x_n$ flow pattern in a Type A network (X, U) and if $(X, \tilde{U})$ and $\tilde{f}$ are defined in the indicated manner then f is ideal in (X, U) if and only if $\tilde{f}$ is ideal in $(X, \tilde{U})$.

Proof. Suppose that f is a feasible flow pattern in (X, U). Let g be another feasible flow pattern in (X, U), having the same value as f, and let $\tilde{f}$ and $\tilde{g}$ be these patterns extended to $(X, \tilde{U})$ by setting $f(v_i) = g(v_i) = 0$ for $i = 2, 3, \ldots, n$. The cost of f relative to (X, U) is the same as that of $\tilde{f}$ relative to $(X, \tilde{U})$ since $f(u) = 0$ for $u \in (\tilde{U} - U)$. Similarly for g and $\tilde{g}$. So if $\tilde{f}$ is ideal in $(X, \tilde{U})$ then f is ideal in (X, U). For otherwise there would be a g in (X, U) with cost smaller than f, and hence $\tilde{g}$ with cost smaller than $\tilde{f}$, which cannot be if $\tilde{f}$ is ideal.

It remains to show that if f is ideal in (X, U) then $\tilde{f}$ is ideal in $(X, \tilde{U})$. The argument here is slightly more involved, since one must consider all feasible flows of a given value in $(X, \tilde{U})$, not merely those such that

$$f(v_2) = f(v_3) = \ldots = f(v_n) = 0 ,$$

in order to determine an ideal flow. Suppose that f is ideal in (X, U), but that $\tilde{f}$ is not ideal in $(X, \tilde{U})$. Let h be ideal in $(X, \tilde{U})$ and have the same value as $\tilde{f}$. Note first that h must be such that $h(v_i) = 1$ for some $i = 2, 3, \ldots, n$. For if h is the extension $\tilde{g}$ of some pattern g in (X, U), since g and $\tilde{g}$ have the same total cost, it may be concluded that g is "cheaper" than f, since f and $\tilde{f}$ have the same cost and $\tilde{g}$ is cheaper than $\tilde{f}$. But this contradicts the assumption that f is ideal. Hence if f is ideal but $\tilde{f}$ is not, then an ideal pattern h in $(X, \tilde{U})$ must be such that $h(v_i) = 1$ for some v_i. For convenience assume that $h(v_2) = 1$. Now $\tilde{f}$ and h are feasible patterns of the same value in $(X, \tilde{U})$. The zero-valued pattern $\tilde{f} - h$ can be decomposed into conformal simple cycle flows $g_1, g_2, \ldots, g_p$. One of these, say g_i, is such that $g_i(v_2) = -1$ since $\tilde{f}(v_2) = 0$ and $h(v_2) = 1$.

From Lemma 2 it is known that $h + g_i$ is feasible. Hence $e(C_i;h) < \infty$, from Lemma 6, where C_i is the simple cycle in $\widetilde{W}$ corresponding to g_i. The cycle C_i consists of some arcs of form $u \in U$ and $u' \in U'$ and of the arc v_2'. For each arc of form u or u' the effective cost relative to h is clearly at most $\max\{a_1(u), a_2(u)\}$. For v_2' on the other hand $e(v_2';h) = -K$ where K is the unit cost $a_1(v_2)$. But $K > \Sigma_U \max\{a_1(u), a_2(u)\}$, and hence clearly $e(C_i;h) < 0$. This contradicts the fact that h is ideal. So if f is ideal in (X, U), $\widetilde{f}$ is ideal in $(X, \widetilde{U})$. This completes the proof.

Note that if f is a feasible $x_1 \rightarrow x_n$ flow pattern in a Type A network (X, U) and if $(X, \widetilde{U})$ and $\widetilde{f}$ are defined in this way then $\widetilde{f}$ is feasible and nonmaximal in $(X, \widetilde{U})$. The latter property follows from the fact that we could augment $\widetilde{f}$ by a unit flow in arc $v_n \cong (x_1, x_n)$ and thus increase the value of flow by one. Moreover $\widetilde{f}$ has the property that, after one iteration of Algorithm 1, every vertex of $(X, \widetilde{U})$ will have a finite label. This follows from the fact that $e(v_j;\widetilde{f}) = K < \infty$ for $i = 2, 3, \ldots, n$, and that v_i directly connects x_1 and x_i. Hence $\widetilde{f}$ meets the requirements for satisfying either Theorem 15 or 21, depending on whether or not it is ideal. These two theorems together with Lemma 12 justify the following general procedure:

PROCEDURE FOR SOLVING PROBLEM C IN TYPE A NETWORKS

(a) Given a feasible $x_1 \rightarrow x_n$ flow pattern in (X, U) define $(X, \widetilde{U})$ and $\widetilde{f}$ in the manner indicated earlier.

(b) Apply Algorithm 1 to $\tilde{f}$ in $(X, \tilde{U})$ and proceed until one of the following occurs:

(i) The algorithm terminates, with $V(y) \leq V(x) + e(\tilde{w};\tilde{f})$ for all $\tilde{w} \in \tilde{W}$ where $\tilde{w} \cong (x, y)$.

(ii) Vertex x_1 is relabeled, so that $V(x_1) < 0$.

(iii) $(n + 1)$ complete iterations have been accomplished.

(c) If case (i) occurs $\tilde{f}$ is ideal; hence (Lemma 22) f is ideal. If case (ii) occurs $A(x_1)$ contains a cycle C such that $e(C;\tilde{f}) < 0$. Replace $\tilde{f}$ by the less costly flow $\tilde{f} + g_c$, where g_c is the elementary flow pattern corresponding to C. If case (iii) occurs, and y is a vertex relabeled during the $(n+1)^{st}$ iteration, then $A(y)$ contains a cycle C with $e(C;\tilde{f}) < 0$. Replace $\tilde{f}$ by $\tilde{f} + g_c$. In either of the last two cases repeat step b with $\tilde{f} + g_c$ in place of $\tilde{f}$. Continue this process until a point is reached such that the outcome of step b is b(i). When this occurs an ideal pattern in $(X, \tilde{U})$ has been found, and also a corresponding ideal pattern in (X, U).

The reason for distinguishing cases (ii) and (iii) in step b of the procedure is simply to shorten the computations whenever $V(x_1) < \infty$ before n+1 iterations have been completed. The procedure can be further shortened by a device essentially the same as that employed in Lemma 7. Once a cycle C has been found such that $e(C;\tilde{f}) < 0$, evaluate Q_w for all $w \in C$ as in Lemma 7. Let $Q = \min_{w \in C} Q_w$. Then $\tilde{f} + Q \cdot g_c$ will be a feasible pattern whose cost is $-Q \cdot e(C;\tilde{f})$ less than that of $\tilde{f}$. This can be established rigorously by a line of reasoning very similar to that employed in the proof of Lemma 7. (The main point in the reasoning is that Q_w is the largest integer such that one can add up to Q_w units of flow to $\tilde{f}(w)$ before the effective cost of w changes.)

REFERENCES

1. R. G. Busacker and P. J. Gowen, "A Procedure For Determining A Family of Minimum-Cost Network Flow Patterns," Operations Research Office, ORO-TP-15, Nov 60.

2. C. Berge, Théorie des Graphes et ses Applications, Dunod, Paris, 1958.

3. D. R. Fulkerson, "An Out-Of-Kilter Method for Minimal Cost Flow Problems," The RAND Corporation, P-1825, revised 5 Apr 60.

4. L. R. Ford, Jr., and D. R. Fulkerson, "Constructing Maximal Dynamic Flows From Static Flows," Opns. Res., 6: 419-33 (1958).

5. M. Iri, "A New Method of Solving Transportation-Network Problems," Opns. Res. Soc. Japan, 3: 27-87 (1960).

6. G. B. Dantzig and D. R. Fulkerson, "Computation of Maximal Flows in Networks," The RAND Corporation, P-677, 1 Apr 55.

7. M. Pollack and W. Wiebenson, "Solutions of the Shortest-Route Problem—A Review," Opns. Res., 8: 224-30 (1960).

8. L. R. Ford, Jr., "Network Flow Theory," The RAND Corporation, P-923, 14 Aug 56.

9. L. R. Ford, Jr., and D. R. Fulkerson, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," The RAND Corporation, RM-1604, 29 Dec 55.

10. W. S. Jewell, "Optimal Flow Through Networks," Massachusetts Institute of Technology, Interim Technical Report No. 8, Jun 58.

GLOSSARY OF PRINCIPAL SYMBOLS

Symbol	Brief Explanation	Page
x, y, z	Individual vertices	7
X	Set of vertices	7
u, v, w	Individual arcs	7
U	Set of arcs	7
$u \cong (x, y)$	"u is directed from x to y"	7
P, C	Path (cycle)	8
u'	"Reverse" of arc u	9
U'	Set of all u' for $u \in U$	9
W	$U \cup U'$	9
f, g, h	Flow patterns	11
f(u), g(u), h(u)	Arc flows	11
$U(x \rightarrow)$, $U(\rightarrow x)$	Set of arcs "leaving" ("entering") x	11
$\Omega_f(x)$	Net output at x relative to f	12
f+g, f-g	Arc-by-arc sum (difference) of f and g	14
g_P, g_C	Elementary path (cycle) flow	15
b(u), c(u)	Lower (upper) bounds on flow	18
$a_1(u)$, $a_2(u)$	Unit cost in direction of u (u')	20
T[f(u)]	Cost of f in u	20
T(f)	Total cost of f: Σ T[f(u)]	20
e(u;f)	Effective cost of arc relative to f	39
e(P;f), e(C;f)	Effective cost of path (cycle) relative to f	40
V(x)	"Label" of x	47
A(x)	Set of approach arcs associated with x	50
A	Set of all approach arcs	50
$(\overline{X}, \overline{U})$	Augmented network for Problem B	58
$(X, \tilde{U})$	Augmented network for Problem C	73

INDEX

Approach
- arc 47
- set 50

Arc 7
- adjacency 8
- approach 47
- end-points of 8
- parallelism 8
- strict adjacency 8
- strict parallelism 8

Cost
- effective 39
- of arc flow 20
- of flow pattern 20
- unit 20

Cycle 9
- effective cost of 40
- flow 15
- saturated 35
- simple 9

Flow 11
- arc 11
- characteristic 39
- elementary cycle 15
- elementary path 15
- feasible arc 18

Flow patterns 11
- conformal 14
- cost of 20
- difference of 14
- elementary 15
- feasible 18
- ideal 20
- sum of 14
- transformable 60
- x→y 14
- value of 14

Graph
- connected 11
- directed 7
- finite 7
- geometric 8

Incidence 8

Lower bound on flow 18

Net output 12

Network 11
- capacitated 18
- capacity of 24
- type A 23
- type B 24
- weighted 20

Path 8
- effective cost of 40
- flow 15
- simple 9
- saturated 35
- x→y 8

Sink 12

Source 12

Upper bound on flow 18

Vertex 7
- initial 7
- intermediate 12
- isolated 8
- net output at 12
- terminal 7